

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest





MAGYAR TUDOMÁNYOS AKADÉMIA  
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

PROGRAMRENDSZER BOOLE-FÜGGVÉNYRENDSZER EGYÜTTES  
EGYSZERÜSÍTÉSÉRE VAGY MINIMALIZÁLÁSÁRA

Irták:

Matavovszky Tibor, dr. Pásztorné Varga Katalin

Tanulmányok 43/1975

A kiadásért felelős

Dr.Arató Máttyás

ISBN 963 311 011 4

Készült az Országos Műszaki Könyvtár és Dokumentációs Központ  
házi sokszorosítójában  
F.v.: Janoch Gyula



## ÖSSZEFOGLALÁS

A programrendszer célja, hogy könnyen kezelhető segédeszközt adjon az elektronikus berendezések tervezése során kapott Boole-függvények kezelésére. Nevezetesen az  $F=\{f_1, f_2, \dots, f_n\}$   $n=1, 2, \dots$  teljesen vagy nem teljesen meghatározott komponens függvényekből álló függvényrendszer együttes minimalizálására vagy egyszerűsítésére. A programrendszer ugyanakkor alkalmas az egyes komponens függvények negáltjának /meghatározott esetben/ vagy részfüggvényeinek / $f_0, f_1, f_\varnothing$ , nem meghatározott esetben/ kiszámolására is. A kitűzött cél érdekében biztosítottuk, hogy a kezelendő  $F=\{f_1, f_2, \dots, f_n\}$  függvényrendszer a megszokott diszjunktív normálformában adható meg. E diszjunktív normálformákat mint adatokat dolgozza fel a programrendszer. A programrendszer működéséhez a komponens függvények formuláin kívül más adat nem kell.

A programrendszer eredményként az egyszerűsítéshez vagy minimalizáláshoz szükséges olyan segédeszközt szolgáltat, amely a gyakorlat szerint a tervező munkájában hatékonyabb segítség, mint egyetlen nem redundáns vagy minimális formula megadása. A kapott eredmény /lásd 2. melléklet 9-10. oldal/:

- az együttes lényeges primimplikánsok listája. A listában feltüntetve, hogy az illető konjunkció mely komponens függvényeknek lényeges primimplikánsa és melyeknek csak primimplikánsa;
- az együttes primimplikánsok listája. A listában feltüntetve, hogy az illető konjunkció mely komponens függvényeknek primimplikánsa. E lista minimalizálás esetén tartalmazza az összes együttes primimplikánst, egyszerűsítés esetén az együttes primimplikánsok egy elegendően bő halmazát;
- a komponens függvényeknek a lényeges primimplikánsok által le nem fedett pontjaiból képzett diszjunkt halmazok listája. E pontoknak a primimplikánsok által való lefedés szempontjából való ekvivalenciaosztályai/[1], 35. oldal/;
- a diszjunkt halmazok primimplikánsokkal való lefedési táblázatát. A táblázatban az egyes diszjunkt halmazoknál feltüntetjük, hogy mely komponens függvényekben nem fedti azt lényeges primimplikáns,

az egyes együttes primimplikánsoknál pedig azt, hogy mely komponens függvényekre nézve primimplikánsok.

A programrendszer szerves része a formulakezelés, amely

- elvégzi a megadott diszjunktív normálformák szintaktikai ellenőrzését;
- eldönti, hogy a komponens függvények mely részfüggvényei adottak és kiszámolja, ha kell az  $f_1, f_0$  részfüggvényeket;
- megvizsgálja az adatként szereplő részfüggvények hibátlanságát /diszjunkt voltát/;
- előállítja a formulák belső gépi reprezentációját.

A programrendszer kezelése - adatmegadás, eredmény kiértékelés - külön erőfeszítést nem követel a felhasználtól. Emellett megoldja az általában csak elvi algoritmus szinten megfogalmazott problémát, a függvényrendszer együttes egyszerűsítését és minimalizálását.



## Обобщение

Цель программной системы – дать легко используемое средство для обработки булевых функций, полученных в этапе проектирования электронных устройств. А именно, для совместной минимизации или упрощения системы функций  $F = \{f_1, f_2, \dots, f_n\}$ , где  $n = 1, 2, \dots$ , состоящей из полностью или неполностью определенных функций. Программная система в то же время способна и для вычисления обратного значения (в определенном случае), или подфункций ( $f_0, f_1, f_\emptyset$  в неопределенном случае) составляющих функций. Для достижения намеченной цели, мы обеспечили систему возможностью задания системы функций в привычной дизъюнктивной нормальной форме. Программная система обрабатывает эти дизъюнктивные нормальные формы, как данные. Для функционирования программной системы, кроме формул составляющих функций, другие данные не требуются. Результат программной системы такое вспомогательное средство для упрощения или минимизации, которое дает – в зависимости от практики – более эффективную помощь ему, чем задание одной избыточной или минимальной формулы. Полученный результат (смотри в приложении № 2, страницы 9–10):

- список совместных существенных простых импликантов. В списке указано, что данная конъюнкция является существенным простым импликантом или только простым импликантом каких составляющих функций;
- список совместных простых импликантов. В списке указано, данная конъюнкция является простым импликантом каких составляющих функций. Этот список при минимизации содержит в себе все совместные простые импликанты, а при упрощении, довольно обширное множество совместных простых импликантов;
- список дизъюнктивных множеств, образованных из точек, не покрытых существенными простыми импликантами составляющих функций. С точки зрения покрытия простыми импликантами, классы эквиваленции этих точек в ([I], стр. 35).
- таблица покрытий дизъюнктивных множеств простыми импликантами. В таблице при каждом дизъюнктивном множестве указано, что он в

каких состояниях функциях не покрыт существенным простым импликантом, а при совместных простых импликантах указано, что они для какой состоящей функции являются простыми импликантами.

Органической частью программной системы является обработка формул, которая

- выполняет синтаксическую проверку заданных дизъюнктивных нормальных форм;
- определяет, какие подфункции составляющих функций заданы, и вычисляет - по необходимости - подфункции
- проверяет безошибочность подфункций, заданных данными (дизъюнктивны ли они)
- вырабатывает внутренне-машинное представление формул.

Использование программной системы - задание данных, оценка результата - не требует особых усилий от пользователя. Кроме этого она решает проблему, поставленную в общем только на уровне принципиального алгоритма-совместное уращение и минимизацию системы функций.



## SUMMARY

The purpose of the program system is to make easier the examination of Boolean functions arising in the planning of electronic devices. In particular it can be used for joint minimization or reduction of systems  $F = \{f_1, f_2, \dots, f_n\}$  consisting of completely or not completely determined functions as components. At the same time the program system can be used to calculate the negation of the component functions /in the case of determined functions/ or subfunctions  $/f_0, f_1, f_2, \dots, f_n$ , i.e. the characteristic function of the three subset, determined by the function in the case of not completely determined functions/. In accordance with the set purpose it was made possible for the function system  $F = \{f_0, \dots, f_n\}$  to be analyzed to be given in disjoint normal form. These disjoint normal forms are processed as data by the program system. To activate the program system the only data needed are the formulas of the component functions.

The output of the program system is more effective help the work of the constructor /as it is shown by practice/ than giving a non redundant or minimal formula; it consists of the following: /see 2nd appendix/.

- The list of the joint essential

primimplicants. In the list it is indicated that which component functions have the conjunction in question as an essential prime implicant and which only as a prime implicant;

- the list of the joint prime implicants.

In the list it is indicated that which of the component functions has the conjunction in question as a prime implicant. In the case of minimization this list contains all the point prime implicants, in the case of reduction it contains a wide enough set of the prime implicants;

- the list of the disjoint sets of the component functions not covered by the essential prime implicants. Equivalence with respect to the covering by prime implicants classes of these points /[1] p. 35/;

- the table of coverings of disjoint sets by prime implicants. In the table it is indicated at the different disjoint sets that

which are the component functions in which they are not covered by essential prime implicants, and at the joint prime implicants that which are the component functions for which they are essential prime implicants.

Integral part of the program system is the formula processing which

- supervises sintactically the given disjunctive normal forms;
- decides that which are the subfunctions of the component functions which are given and determines if necessary the subfunctions  $f_1, f_0$ ;
- tests the correctness /disjointness/ of the subfunctions given as data;
- produces the inner machine representation of the formulas.

To work with the program system there is not needed a special effort like data transformation, interpretation of the result. Moreover it solves the problem formulated usually only on the level of abstract algorithm: joint minimization and reduction of function systems.



## 1. MINIMALIZÁLÓ ÉS EGYSZERÜSÍTŐ PROGRAMRENDSZER

### 1.1 A program feladata

Diszjunktív normálformában /DNF/ megadott egyetlen teljesen meghatározott vagy nem teljesen meghatározott Boole-függvénynek, valamint több nem feltétlenül teljesen meghatározott függvénynek /függvényrendszernek/ az előforduló változók száma szerinti együttes minimalizálása vagy egyszerűsítése ill. a minimalizáláshoz vagy egyszerűsítéshez szükséges adatok szolgáltatása. Ezek az adatok a teljes primimplikáns lista, a lényeges primimplikánsok listája, a lényeges primimplikánsok által le nem fedett diszjunkt tartományok listája és lefedési táblázat e diszjunkt halmazokra. Az  $F$  függvényrendszert komponens függvényein keresztül  $F = \{f_1, f_2, \dots, f_m\}$  adjuk meg. A komponens függvényeket pedig az  $f_{11}$  és  $f_{10}$  függvényekkel kell megadni az algoritmus számára.

$$f_{11} = \begin{cases} 1 & \text{ha } f_1 = 1 \\ 0 & \text{egyébként} \end{cases}$$

$$f_{10} = \begin{cases} 1 & \text{ha } f_1 = 0 \\ 0 & \text{egyébként} \end{cases}$$

A teljesen meghatározott  $f$  függvényre  $f_1 = f$  és  $f_0 = \bar{f}$ .

Szokás a nem teljesen meghatározott függvényt  $f_1$  és  $f_\phi$ -val is megadni, ahol

$$f_\phi = \begin{cases} 1 & \text{ha } f \text{ nem meghatározott} \\ 0 & \text{egyébként.} \end{cases}$$

$f_\phi$ -t szokás a meghatározatlansági tartomány karakterisztikus függvényének is nevezni. További elnevezések és jelölések:  $f_1 = f$ ; az  $f$  alsó határa.  $\hat{f} = f_1 \vee f_\phi$ ; az  $f$  függvény felső határa.

## 1.2 A program szerkezete

A minimalizáló és egyszerűsítő program egyaránt három fő részből és egy kisegítő részből áll. Ezek a

- beolvasó program
- minimalizáló algoritmus vagy egyszerűsítő algoritmus
- az algoritmus eredményének kiértékelése és kiírása szélesnyomtatón
- bitműveleteket végző FORTRAN rutinok.

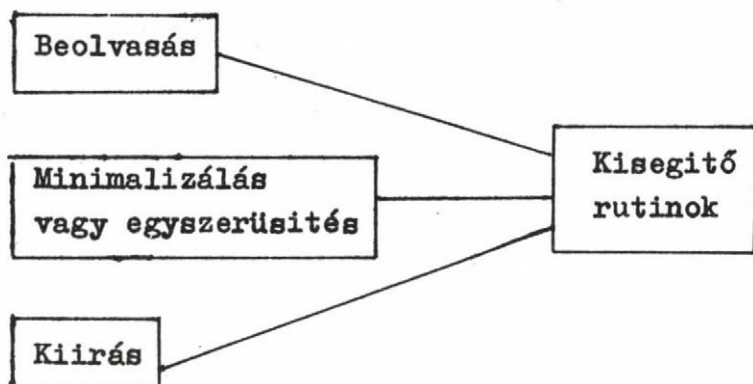
A beolvasó program a függvény DNF-ját beolvassa, és követési mátrix technikával [2] szintaktikailag ellenőrzi azt. Hiba esetén hibajelzést ad, és a szintaktikai vizsgálat után nem folytatja a munkát. Ha a formula helyes volt, a függvény DNF-jának egy olyan belső gépi reprezentációját állítja elő, amely a minimalizáló algoritmus bemenő adata.

A minimalizáló ill. az egyszerűsítő algoritmus a gráf módszer [1] alapján felírja az összes primimplikánst vagy együttes primimplikánst ill. egy egyszerűsítést lehetővé tevő primimplikáns vagy együttes primimplikánshalmazt, és megjelöli a lényegeseket. A függvény pontjait a primimplikánsokkal való lefedés alapján ekvivalenciaosztályokba osztja /diszjunkt halmazok/, és megjelöli azokat a diszjunkt halmazokat, amelyeket lefed lényeges primimplikáns.

A kiértékelő és kiíró programrész kinyomtatja a minimalizált függvények neveit, a változók számát, a teljes primimplikáns listát, megjelölve a lényeges primimplikánsokat és egy lefedési táblázatot a lényeges primimplikánsok által le nem fedett diszjunkt tartományokra és a nem lényeges primimplikánsokra.

A kisegítő rutinok, amelyeket az előző három programrész használ fel, a gépi szó tartalmának bitenkénti kezelését teszik lehetővé. Ezek a rutinok egy vagy két gépi szóra elvégzik bitenként a szokásos logikai műveleteket /negáció, diszjunkció, konjunkció stb./

A program sémája:







## 2. A PROGRAM RÉSZLETES LEÍRÁSA

### 2.1 A beolvasó program

A beolvasó program alapfunkciója a DNF szintaktikai ellenőrzése és gépi reprezentációjának előkészítése.

A beolvasó program lyukkártyáról beolvassa az első  $\pi$  jelig a függvények olyan formuláit, amelyek  $\wedge, \vee, \neg, =, -$ -en kívül más jeleket nem tartalmaznak. E formulákat szintaktikailag ellenőrzi. A diszjunktív normálformában megadott függvényekben szereplő konjunkciók belső /helyértékes/ reprezentációját elkészíti. A helyértékes reprezentáció azt jelenti, hogy az egyes változóknak  $2^i$  alaku belső neveket feleltetünk meg / $i=0,1,\dots,n-1$ ;  $n$  a változók száma/. E reprezentáció szerint egy egész típusu szóban /a CDC 3300 gépen/ 23 változót helyezhetünk el. Egy konjunkció belső reprezentációjához két egész típusu szót használunk fel az első komponensben a konjunkcióban negálatlanul szereplő változóknak megfelelő  $2^i$  alaku számok összege, a második komponensben a konjunkcióban előforduló változóknak megfelelő  $2^i$  alaku számok összege szerepel.

P1. legyen a változók száma 4. A változók:  $x_1, x_2, y, z$ .

A helyértékes reprezentációban  $x_1 \rightarrow 1, x_2 \rightarrow 2, y \rightarrow 4, z \rightarrow 8$ .

Az  $x_1 \bar{y}$  konjunkció belső reprezentációja:

|      |              |
|------|--------------|
| 0001 | 1. komponens |
| 0101 | 2. komponens |

Tekintettel arra, hogy a mintermekben minden változó szerepel egy mintermnek megfelelő 2. komponens első  $n$  helyértékén csupa 1-es áll. Ennek megfelelően a mintermek belső reprezentációjában a 2. komponens elhagyható. Példánkban 0110 sorozat az  $\bar{x}_1 x_2 y \bar{z}$  mintermet reprezentálhatja.

A beolvasó program szubrutinjai:

|            |         |
|------------|---------|
| BEOLV      | SINT    |
| ALAKIT     | ATIR1   |
| VAL        | FREPR   |
| BLOCK DATA | VALTNIV |
| SZERK      | ATIRT   |
| LISTAK     | ATIRG   |
| KONJ       |         |

A szubrutinok kapcsolódása az 1. ábrán látható.

## 2.2 A program által végrehajtandó feladatok

### A/ Boole-függvény ill. függvények beolvasása

#### Alaki követelmények:

A függvény diszjunktív normálformájának szokásos alakját minimális mértékben kell csak átalakítani /linearizálni/, hogy a beolvasó számára érthető legyen.

A formulában felhasznált jelek a következők:

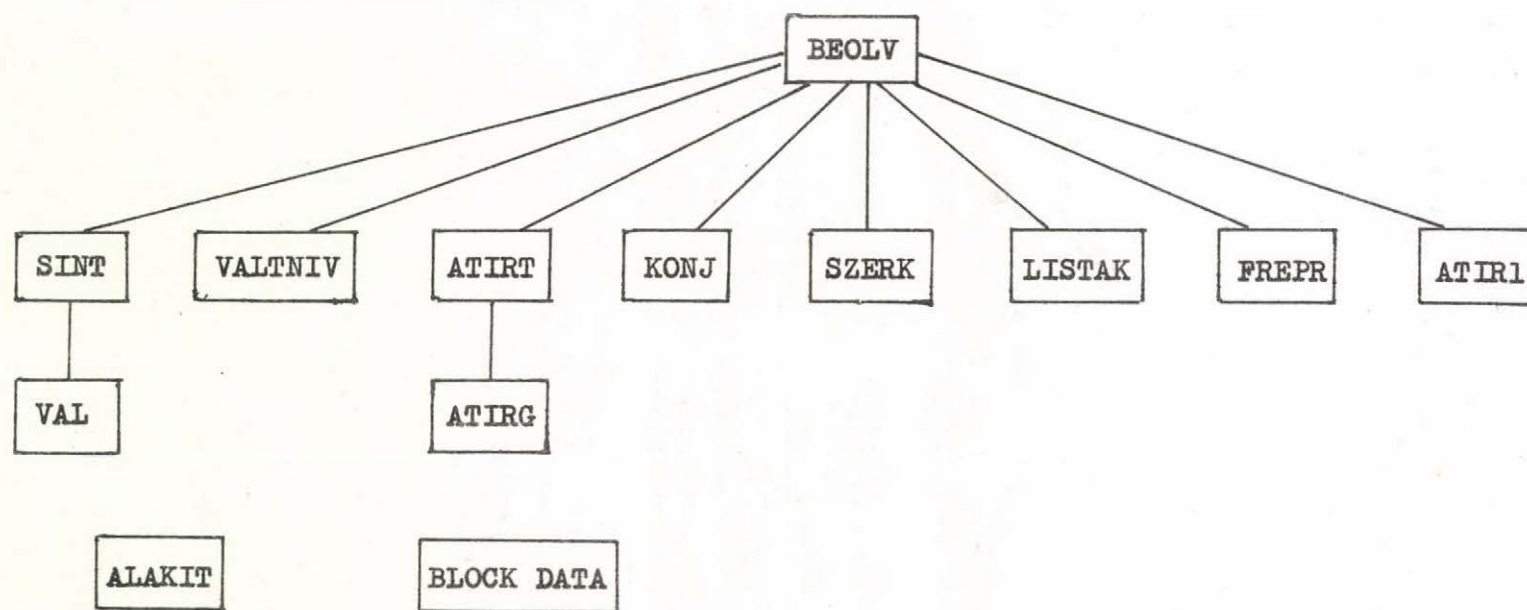
|                      |                                      |
|----------------------|--------------------------------------|
| diszjunkció jele:    | +                                    |
| konjunkció jele:     | .                                    |
| negáció jele:        | -                                    |
| "függvény vége" jel: | ,                                    |
| "adat vége" jel:     | ⌘                                    |
| változónév           | betűvel kezdődő legfeljebb négyjegyű |
| függvénynév          | alfanumerikus karaktersorozat lehet. |

Pl. ha  $f_1 = a_1 \wedge b_1 \vee \bar{c} \vee \bar{a}_1$  és  $hx_2 = x \vee \bar{y}_2$  formulákat kell megadni a program bemenő adataként, akkor a formulák bemenő adattá való átírásának eredménye:  $f1=a1.b1.-c+-a1$ ,  $hx2=x+-y2$  ⌘. A formula felírásában a space jeleknek nincs jelentőségük.

A beolvasó program a beolvasott függvények DNF-i belső reprezen-



A BEOLVASÓ PROGRAM SZUBRUTINJAINAK KAPCSOLÓDÁSA



1. ábra

tációiból a minimalizáló program bemenő adatát elkészíti az alábbi esetekben:

a/ Egyetlen teljesen meghatározott függvény esetén

Ekkor az egy teljesen meghatározott függvényt minimalizáló speciális algoritmus számára előállítja a függvény kitüntetett diszjunktív normálformáját a függvény valamely tetszőleges DNF-ja alapján.

A bemenő adat a függvény tetszőleges DNF-ja, azaz

$$f_{10} = a.b5.-d6+d6.-a \text{ \#}$$

b/ Nem teljesen meghatározott függvény vagy függvényrendszer esetén

A függvényrendszert /egyetlen függvényből állót is/ minimalizáló program bemenő adata a komponens függvényeket megadó  $f_{11}$ ,  $f_{10}$  függvények kitüntetett diszjunktív normálformái. E függvények kötelező jelölése a beolvasó program számára  $f_{11}=f_1$ ,  $f_{10}=-f_1$ . Az említett bemenő adatot a beolvasó program a következő három esetben tudja előállítani.

b-1. Megadjuk egyes komponens függvényekhez tartozó  $f_{11}$ ,  $f_{10}$  tetszőleges DNF-it bármilyen sorrendben.

$$P_1. \text{ ha } fa_1 = d \wedge \bar{a} \wedge \bar{b} \wedge \bar{c} \vee \bar{a} \wedge b \wedge c \wedge d,$$

$$fa_0 = a \wedge b \wedge \bar{c} \wedge \bar{d} \vee \bar{a} \wedge \bar{c} \wedge b \wedge d,$$

$$fb_1 = d \wedge \bar{a} \wedge \bar{b} \wedge \bar{c}, \quad fb_0 = b \wedge \bar{a} \wedge \bar{c} \wedge \bar{d} \vee a \wedge c \wedge \bar{b} \wedge \bar{d}$$

az egyszerűsítendő függvények, akkor a program egy bemenő adata a következő lehet:

$$fa = d.-a.-b.-c+.-a.b.c.d, \quad -fa = a.b.-c.-d+.-a.-c.b.d,$$

$$fb = d.-a.-b.-c, \quad -fb = b.-a.-c.-d.+a.c.-b.-d \text{ \#}$$

b-2. Az  $f_{11}$ ,  $f_{10}$ -al meg nem adott komponens függvények mind teljesen meghatározottak, és  $f_1$  vagy  $\bar{f}_1$  közül legalább az egyik ismert. Ebben az esetben meg kell adni az ismert függvények tetszőleges DNF-it tetszőleges sorrendben, és utolsó függvényként egy, a már felsorolt függvényekben szereplő változók konjunkcióját, azaz egy  $S = \bigvee_{i=1}^K x_i$  formulát, ahol S olyan változónév, amely nem szerepel komponens függvény neveként, K pedig a komponens függvényekben

előforduló változók száma.  $S \rightarrow \{i\}$

Pl.  $F = \{f_1, f_2, f_3, f_4\}$  esetén, ha adott  $f_1, \bar{f}_1, f_2, \bar{f}_3, f_{4,1}, f_{4,0}$ , akkor a bemenő adatok

$$-f_4 = -c.p.t + -t.-c.-a + -p.-c,$$

$$f_2 = a + b.p + -b.t + p.t, \quad -f_3 = a.c.p. - n + a.-c,$$

$$f_1 = a.-t.b + -a.b, \quad f_4 = c.-t, \quad -f_1 = -b + a.t,$$

$$g = a.b.c.p.t. - n$$

b-3. Az  $f_1, \bar{f}_1$  vagy  $f_{11}, f_{10}$  alakban pontosan meg nem adott komponens függvények  $g$  meghatározatlansági tartománya közös. Ekkor meg kell adni az ismert függvények tetszőleges DNF-it tetszőleges sorrendben, és utolsó függvényként a  $g$  függvény DNF-ját, de  $-g = \dots$  alakban.

Pl.  $F = \{f_1, f_2, f_3, f_4, f_5\}$ , ahol  $f_1$  teljesen meghatározott és  $f_1, \bar{f}_1$  adott,  $f_2$  nem teljesen meghatározott, de  $f_{21}, f_{20}$  adott,  $f_3, f_4, f_5$  meghatározatlansági tartománya, amelynek karakterisztikus függvénye  $g$ , közös és adottak:  $g, f_{31}, f_{41}, f_{50}$ . A beolvasó program egy lehetséges bemenő adata:

$$f_1 = d.-e.f + -d.f, \quad -f_1 = -f + d.e,$$

$$f_2 = f.-e, \quad -f_2 = -f.d.e + -e.-f.-a + -d.-f,$$

$$f_3 = d.e.-f.a.z + d.e.-f.-z,$$

$$f_4 = -d.-e.-f.a + -d.e.-f.z.a,$$

$$-f_5 = d.-f.-e.z.a + d.-e.f,$$

$$-g = d.e.f + -d.-e.f + -d.e.f$$

Az adatokat lyukkártyán kell rögzíteni. A lyukkártya 1-80 oszlopai tetszés szerint felhasználhatók, mivel a space jelnek nincs jelentősége /lásd 2. ábra/. Uj sorra való át-térés mindig elválasztó jelnél =, +, ., ", " történhet.

Folytatás jel nincs.

Méretre vonatkozó előírások:

Egy függvényen belül legfeljebb 23 változó szerepelhet. Az egyszerre beolvasásra kerülő karakterek száma legfeljebb 5000 lehet.



## B/ A beolvasott karaktersorozat szintaktikai vizsgálata

Vizsgálja a formulák helyességét. A megadott alaki követelmények alapján egy szimbólumnak számít minden beolvasott karakternél vagy karaktersorozatnál megvizsgálja, hogy az illető szimbólum következhet-e az előző után. Összeállítottuk a követési mátrixot, amelynek alapján a szintaktikai vizsgálat elvégezhető /lásd 2. ábrát/.

| i-edik karakter | i+1-edik karakter |   |   |   |   |   |   |   |   |   |
|-----------------|-------------------|---|---|---|---|---|---|---|---|---|
|                 | -                 | . | + | ( | ) | b | s | , | = | ⌘ |
| -               |                   |   |   | 1 |   | 1 |   |   |   |   |
| .               | 1                 |   |   | 1 |   | 1 |   |   |   |   |
| +               | 1                 |   |   | 1 |   | 1 |   |   |   |   |
| (               | 1                 |   |   | 1 |   | 1 |   |   |   |   |
| )               |                   | 1 | 1 |   | 1 |   |   | 1 |   | 1 |
| b               |                   | 1 | 1 |   | 1 | 1 | 1 | 1 | 1 | 1 |
| s               |                   | 1 | 1 |   | 1 | 1 | 1 | 1 | 1 | 1 |
| ,               | 1                 |   |   |   |   | 1 |   |   |   |   |
| =               | 1                 |   |   | 1 |   | 1 |   |   |   |   |
| ⌘               |                   |   |   |   |   |   |   |   |   |   |

b - betű

s - számjegy

2. ábra

Az esetleges hibaüzenetek értelmezését megkönnyíti, hogy egy kártya beolvasása után annak tartalma változatlan formában kiírásra kerül. A szintaktikai vizsgálat formulánként, kártyákra bontva történik.

### Hibaüzenetek

- Ha két karakter szekvenciája a követési táblázat alapján hibás, akkor kiírásra kerül az
  - NEM MEGENGEDETT KARAKTEREK KÖVETIK EGYMÁST üzenet, továbbá
  - a két karakter és az MAL tömb azon elemének indexe, ahová

az első karakter elhelyezésre került a formula szintaktikai vizsgálatával párhuzamosan folyamatban lévő belső reprezentáció előállítása során.

2. Ha a változónév négynél több karakterből áll, akkor minden egyes többletkarakter felfedezése alkalmával kiírásra kerül  
a/ EGY VALTOZONEV NEGYNEL TOBB KARAKTERT TARTALMAZ üzenet, továbbá  
b/ a karakter;  
c/ a név első négy karakteréből kialakított belső reprezentációt tartalmazó MAL tömb elemének indexe;  
d/ a karakter sorszáma változónéven belül.
3. Ha két formula-elválasztó jel között egynél több az egyenlőségjel, akkor a hibaüzenet:  
a/ EGYENLOSEGJEL FELESLEGES  
b/ a karakter;  
c/ a belső reprezentációt tartalmazó MAL tömb azon elemének indexe, amelybe az = jel elhelyezésre került;  
d/ a szám, amely megmutatja, hogy hányadik az egyenlőségjel az aktuális formulában.
4. Ha a formulában felesleges végzárójelet találunk, akkor a hibaüzenet:  
a/ TOBB, A VEGZAROJEL  
b/ a karakter;  
c/ a belső reprezentációt tartalmazó MAL tömb azon elemének indexe, ahová a ")" elhelyezésre került;  
d/ a többlet végzárójelek száma negatív előjellel.
5. Ha egy formula vizsgálata befejeződött, és benne a kezdő- és végzárójelek száma nem egyezik meg, akkor a hibaüzenet:  
a/ A FUGGVENYBEN A KEZDO ES VEGZAROJELEK SZAMA KULONBOZIK  
b/ a formula határoló karaktere;  
c/ az MAL tömb azon elemének indexe, ahová a határoló jel került;  
c/ a kezdő és végzárójelek különbsége.

Tulcsordulást jelző hibaüzenetek:

PONTOK SZAMA GE 500

500-NAL TOEBB 1-PONT ATIRT-BAN



C/ A formulák egy belső reprezentációjának létrehozása. /A szintaktikai vizsgálattal párhuzamosan történik./

D/ A Boole-függvény beolvasott formulájával azonos jelsorozat ki-nyomtatása. /A beolvasás után közvetlenül történik./

Szintaktikai hiba esetén a hibaüzenet után tovább fut a program és a teljes formulahalmazra elvégzi a szintaktikai vizsgálatot, majd a "SZINTAKTIKAI HIBAK" kiírása után megáll. A többi hibaüzenet esetén a program végrehajtása azonnal felfüggesztődik.

### 2.3 A feladatsorozat végrehajtásának módja

#### Karaktersorozat beolvasása

A Boole-függvény vagy függvények beolvasásra kerülő formulájának megfelelő karaktersorozatot folyamatosan vagy betűközök közbeiktatásával kártyára lyukasztjuk. Az egyes formulákat a (,) választja el, az utolsó formula után pedig a (\*) jel kerül.

A beolvasás a karaktereket egyenként beolvassa a MA(I) tömbbe /A1 formátummal./

#### A szintaktikai vizsgálat megvalósítása

3 db 10x10-es mátrix felhasználásával történik, amelyek közül kettő lényegében követési mátrix /3,4 ábra/. A követési mátrix sorait és a lehetséges karaktertipusokat egy-egyértelműen megfeleltetjük egymásnak. Az egyes sorokban azok a karaktertipusok találhatók, amelyek a sornak megfeleltetett karakter után következhetnek. A harmadik mátrix maszkolásra szolgál /5. ábra/, mivel e mátrixok használatához a mátrix egy sorát egyetlen félszóba helyezzük el.

E mátrixok beolvasása a FORTRAN jelkészlettel együtt /6. ábra/ a BLOCK DATA szegmensben történik.



## Változónév belső reprezentációjának előállítása

A változó elhelyezésére szolgáló tömbbelembe elhelyezzük az első karakter belső reprezentációját. A többi karakter esetén a már bennlévő értéket  $2^6$ -nal megszorozzuk és az új karakter belső reprezentációját hozzáadjuk.

## Formula kinyomtatás

A belső reprezentációból a karakterszámtól ( $K$ ) függően  $2^{6(K-1)}$ -el való osztás segítségével előállítjuk az első karakter belső reprezentációját és az ennek megfelelő MF tömbbeli karaktert írjuk a kinyomtatásra kerülő tömb következő félszavába. A belső reprezentációból töröljük a karakter belső reprezentációját, és az eljárást ismételjük  $K$ -nál 1-gyel kisebb értékkel, amíg a kitevő  $(6(K-1))$  értéke 0-ra nem csökken.

## 2.4 A megvalósított program ismertetése

A BEOLV program által használt változók és tömbök definíciója.

### Munkaváltozók:

C L:           Ha  $L=0$  a szintaktikai vizsgálat nem talált,  
              ha  $L=1$  talált szintaktikai hibát.  
/A (C) jel a továbbiakban a COMMON-beli területet jelent./

### Tömbök:

C MF (45):     A FORTRAN jelkészlet tárolása /5. ábra/  
C MA (I):     A formulában szereplő karakterek elhelyezésére szolgáló tömb.

### A SINT szubrutin

A beolvasandó függvények szintaktikai ellenőrzését végzi.

### Munkaváltozók:

|        |   |
|--------|---|
| J1:    | A karakterkeresés során megtalált karakter MF tömbbeli belső reprezentációja.   |
| (C) M: | MA1 tömb feltöltésénél az aktuális sorszám tárolása. A ciklus befejezése után az MA1 tömbben feltöltött szók számát tartalmazza.                        |
| MVEKT: | A karaktertípushoz a MATRIX mezőben rendelt sorszám.  |
| (C) L: | Ha szintaktikai hiba van értéke 1, különben 0.  |
| IKEZD: | Két elválasztó jel (,) között a zárójelek számlálására szolgáló változó. Értéke 1-gyel nő, ha kezdő, és 1-gyel csökken, ha záró zárójelet talált a gép. |
| MIE:   | Két vessző (,) jel között az = jelek számlálására szolgál.  |
| LZ:    | Ha a szekvencia megengedett, akkor az értéke 0, egyébként 1.  |
| MENT:  | A beolvasott kártya utolsó karakterének tárolása.   |
| KBK:   | Változó vagy függvénynév karaktereinek számlálása.  |
| IE:    | Értéke 1 vagy 0, aszerint, hogy a vizsgált változó függvény- vagy változónév.   |
| Il:    | Hibajelzéshez szükséges sorszám, ha két egymást követő karaktert akarunk kinyomtatni.   |
| KM:    | A beolvasott kártya utolsó elemének indexe az MA tömbben.   |

### Tömbök:

|               |  |
|---------------|--|
| (C) KAR (10): | A karakterhez rendelt $2^i$ alaku belső reprezentáció, $KAR(k) = 2^{k-1}$ ( $k=1, \dots, 10$ ).  |
| IV1 (10):     | Összeférhetőségi vektorok tömbje. Ha egy s típusu karaktert $k_1, \dots, k_\ell$ ( $0 \leq \ell \leq 10$ ) tipusszámú karakterek követhetik (, és = jel között), akkor $IV1(s) = 2^{k_1-1} + \dots + 2^{k_\ell-1}$ . |

IVO(10):           Összeférhetőségi vektorok tömbje (=) jel után és  
                  (,) előtt. Szerkezete azonos IV1(10)-ével.

(C)MAL(I):         Függvény vagy változónév belső reprezentációja  
                  vagy egyéb karakterek esetén maga a karakter belső  
                  reprezentációja.

VAL (KA,KC,I) szubrutin

Eldönti, hogy a beolvasott karakter az MVEKT alapján, következhet-e  
az előző karakter után.

Munkaváltozók:

KAM:               MVEKT megváltozott értéke

IM:                A törlést végző ciklusváltozó végértéke

Formális paraméterek:

MVEKT-nek megfelel KA  
LZ-nek megfelel KC  
NUM-nak megfelel I

SZERK szubrutin

Feladat:

A beolvasott függvények szerkezetének vizsgálata alapján az IR (K,J)  
kitöltése. Zárójeles kifejezés esetén K=1,2,3,4,5,6, DNF esetén  
K=4,5,6.

Munkaváltozók:

(C)IS:            A függvények száma

M3:               A soron következő függvéynév tárolása

M4:               A függvény első karakterének belső reprezentációja

K:                A beolvasott függvényekben található zárójelek szá-  
                  ma



- L1:           Értéke 1, ha a beolvasott függvények mindegyike diszjunktív normálforma.
- M6:           Az IR tömb aktuális sorszáma
- M5:           A vizsgálat alatt álló zárójeles függvény sorszáma
- IT:           A függvény típusát jellemző számérték kialakítása. Értéke  $K$  ( $0 < K < 4$ ), ha a függvény  $K$  mélységben tartalmaz zárójelet.
- J2:           Egy függvény azonos típusu függvényeinek számlálása
- K2:           Az összetartozó zárójelpárokat kiválogató ciklusváltozó végértéke
- J3:           A vizsgálat során talált belső függvények száma
- J1:           Az első belső függvény sorszáma
- M1:           Egy függvény belső függvényeinek számlálása

Tömbök:

(C)IR(6,I)

1. sor:       IR(1,a) a beolvasott függvényhez tartozó belső függvények száma, IR(1,a+1) belső függvénynél a beolvasott függvény sorszáma
2. sor:       IR(2,a) a zárójelmélység a beolvasott függvényben. Az egy függvényen belüli azonos típusu belső függvények sorszáma
3. sor:       Üres  
Tipusszám = k, jelentése: a függvény k-1 mélységben tartalmaz zárójelet.
4. sor:       A beolvasott függvény név első belső reprezentációja  
A függvény neve a karakterek belső reprezentációjából kialakítva
5. sor:       A függvény /belső is/ első karakterének M1 tömbbeli indexe
6. sor:       A függvény utolsó karakterének M1 tömbbeli indexe.

## LISTAK szubrutin

### Feladat:

A beolvasott L darab függvényre az NF(7,L) tömb első öt sorának kitöltése.

A változólista elkészítése.

A függvénynevek és az egyes függvényekhez tartozó változólisták kinyomtatása.

### Munkaváltozók:

|      |  |
|------|--|
| M1:  | A függvénylista aktuális sorszáma                                      |
| M2:  | A változólista elemeinek folyamatos számlálása                         |
| M22: | A változólista elemeinek számlálása egy függvényen belül               |
| N3K: | A függvénylistára utoljára felvett függvény első változójának sorszáma |
| M4:  | A függvény belső reprezentációjához tartozó kitevő                     |
| M5:  | A változó belső reprezentációjához tartozó kitevő                      |
| M10: | A függvény első karakterének vizsgálatához a sorszám tárolása          |
| IC1  | A változólistát kinyomtató ciklusváltozó kezdő és végértéke.           |
| IC2  |  |

### Tömbök:

(C) NF (7,I)

|         |  |
|---------|--|
| 1. sor: | A függvénynév első belső reprezentációja       |
| 2. sor: | A függvény belső reprezentációja               |
| 3. sor: | Értéke 1, ha a függvény negálatlan, különben 0 |
| 4. sor: | Változólista kezdete                           |
| 5. sor: | Változólista vége                              |
| 6. sor: | Konjunkciólista kezdete                        |

7. sor: Konjunkciólista vége  
(C) JV(I) Változólista

KONJ szubrutin

Feladat:

Diszjunktív normálformában megadott függvények konjunkciólistájának elkészítése.

A konjunkciólista és a függvéynév kinyomtatása.

Munkaváltozók:

M7: A konjunkciólista aktuális sorszáma  
MDISZ: Értéke 1, ha a vizsgált változó plusz (+) jel, egyébként 0.  
IC1 A konjunkciólistát előállító ciklusváltozójának kezdő (IC1) és végértéke (IC2) karaktervizsgálatnál.  
IC2  
IC7: A függvény változólistája első elemének címe  
IC8: A függvény változólistája utolsó elemének címe

Tömbök:

K2(I): A konjunkcióban előforduló változók belső reprezentációinak összege  
K1(I): A konjunkcióban negátlanul előforduló változók belső reprezentációinak összege

ALAKIT szubrutin

Feladat:

Egy első belső reprezentációval megadott függvény vagy változónév visszaalakítása  $K$  ( $1 \leq k \leq 4$ ) elemű karaktersorozattá. A kialakított karaktersorozatot az MA(I) tömbbe helyezi el.



### Munkaváltozók:

- (C)K1: A változó vagy függvénynév első belső reprezentációját visszaalakító ciklus kezdő értéke =  $5-x/x$  a név karaktereinek száma/.
- MMA2: Az első belső reprezentációból leválasztott karaktersorszám kialakítása.
- MKARN: A leválasztott karaktersorszám tárolása.

### ATIR1 szubrutin

Egy teljesen meghatározott függvény konjunkciói alapján felírja a függvény mintermjeit. Ha a mintermek száma eléri az 500-at, kinyomtatódik a - PONTOK SZÁMA GE 500 - hibajelző üzenet.

Paraméterátadás a közös területen történik.

### Közös területek:

/KONJ/ K1(I), K2(I) - konjunkciólista  
/BOOLE/ MA(6,I)-MA(1,I)-be kerül a mintermlista  
IPE - a változók száma  
IS - a feldolgozandó függvények száma  
/LIST/ NF(7,23) - a függvényekre vonatkozó adatok.

### ATIRT szubrutin

A komponens függvények  $f_{i1}, f_{i0}$  alaku megadásánál e függvények konjunkció listája alapján megadja  $f_{i1}$  és  $f_{i0}$  minterm listáját. Ha a komponens függvények megadása nem  $f_{i1}, f_{i0}$  alaku, akkor a feldolgozás az ATIRG szubrutinban folytatódik.

Ha az  $f_{i1}$  mintermjeinek együttes száma eléri az 500-at, akkor a - 500-NAL TOEBB 1-PONT ATIRT-BAN - hibaüzenet, ha az  $f_{i0}$  mintermjeinek együttes száma eléri az 500-at, akkor a - 500-NAL TOEBB 0-PONT ATIRT-BAN - hibaüzenet nyomtatódik ki.

Paraméterátadás a közös területen történik.

### Közös területek:

/KONJ/ K1(I), K2(I) - konjunkciólista

/LIST/ NF(7,23) - a függvényekre vonatkozó adatok

/BOOLE/ MA(6,I), MB(2,I) - MA(1,I)-be kerülnek az  $f_{i1}$  mintermjei, MA(2,I)-be az I-edik minterm hovátartozását mutató index. MB(1,I)-be kerülnek az  $f_{i0}$  mintermjei, MB(2,I)-be az I-edik minterm hovátartozását mutató index.

IS - beolvasott függvények száma

IP -  $f_{i1}$  formulák száma

IO -  $f_{i0}$  formulák száma

### ATIRG szubrutin

Ha a komponensfüggvények között szerepel  $f_{i1,g}$  és /vagy/  $f_{j0,g}$  /g közös/ alakban megadott függvény vagy  $f$  ill.  $\bar{f}$ -tal megadott teljesen meghatározott komponens függvény, akkor a szubrutin a megadott függvények mintermjei alapján előállítja az összes  $f_{i1}, f_{i0}$  minterm listáját.

Paraméterátadás a közös területen történik.

Közös terület: megegyezik az ATIRT közös területével.

### VALTNIV szubrutin

A beolvasott függvényekben előforduló változókból listát készít, és az egyes függvények változólistáját erre a közös változólistára változtatja.

Paraméterátadás a közös területen történik.

### Közös terület:

IV - változók száma

IS - függvények száma

/LIST/ NF(7,23) - a beolvasott függvényekre vonatkozó adatok

JV(I) - az egyes függvények változóinak listái vannak benne, és a közös változólista kerül bele.

### FREPR szubrutin

Azon függvényekhez, amelyeknek függvényazonosítója csak a "-" jelében tér el egymástól, azonos  $2^i$  alakú indexet rendel.

Paraméterátadás a közös területen történik.

#### Közös terület:

LF - a különböző nevű függvények száma

LS - a beolvasott függvények száma

/LIST/ NF(7,23) - a függvényekre vonatkozó adatok. NF(2,I)-be kerül az I-edik beolvasott függvény indexe.



A követési mátrix: a vizsgált és az előző karakter az (=) jobb oldalán szerepel:

|                |   | I  | IVO(I) |   |   |   |   |   |   |   |  |   |     |
|----------------|---|----|--------|---|---|---|---|---|---|---|--|---|-----|
| negáció        | - | 1  |        |   |   | 1 |   | 1 |   |   |  |   | 80  |
| konjunkció     | . | 2  | 1      |   |   | 1 |   | 1 |   |   |  |   | 592 |
| diszjunkció    | + | 3  | 1      |   |   | 1 |   | 1 |   |   |  |   | 592 |
| kezdő zárójel  | ( | 4  | 1      |   |   | 1 |   | 1 |   |   |  |   | 592 |
| végzárójel     | ) | 5  |        | 1 | 1 |   | 1 |   |   | 1 |  |   | 420 |
| betű           | x | 6  |        | 1 | 1 |   | 1 | 1 | 1 | 1 |  | 1 | 445 |
| szám           | 2 | 7  |        | 1 | 1 |   | 1 | 1 | 1 | 1 |  | 1 | 445 |
| függvény vége  | , | 8  | 1      |   |   |   |   | 1 |   |   |  |   | 528 |
| egyenlőség     | = | 9  | 1      |   |   | 1 |   | 1 |   |   |  |   | 592 |
| beolvasás vége | ≠ | 10 |        |   |   |   |   |   |   |   |  |   | 0   |

3. ábra

A követési mátrix, ha a vizsgált és az előző karakter az (=) bal oldalán szerepel:

|                |   | I  | IVl(I) |  |  |  |   |   |  |   |  |     |
|----------------|---|----|--------|--|--|--|---|---|--|---|--|-----|
| negáció        | - | 1  |        |  |  |  | 1 |   |  |   |  | 16  |
| Konjunkció     | . | 2  |        |  |  |  |   |   |  |   |  | 0   |
| diszjunkció    | + | 3  |        |  |  |  |   |   |  |   |  | 0   |
| kezdő zárójel  | ( | 4  |        |  |  |  |   |   |  |   |  | 0   |
| záró zárójel   | ) | 5  |        |  |  |  |   |   |  |   |  | 0   |
| betű           | x | 6  |        |  |  |  | 1 | 1 |  | 1 |  | 26  |
| szám           | 2 | 7  |        |  |  |  | 1 | 1 |  | 1 |  | 26  |
| függvény vége  | , | 8  | 1      |  |  |  | 1 |   |  |   |  | 528 |
| egyenlőség     | = | 9  |        |  |  |  |   |   |  |   |  |     |
| beolvasás vége | ≡ | 10 |        |  |  |  |   |   |  |   |  |     |

4. ábra

A karaktertipusokhoz rendelt belső jelzőszámok /maszkolás/:

|                |   | I  | KAR(I) |   |   |   |   |   |   |   |   |   | : |     |
|----------------|---|----|--------|---|---|---|---|---|---|---|---|---|---|-----|
| negáció        | - | 1  | 1      |   |   |   |   |   |   |   |   |   |   | 512 |
| konjunkció     | . | 2  |        | 1 |   |   |   |   |   |   |   |   |   | 256 |
| diszjunkció    | + | 3  |        |   | 1 |   |   |   |   |   |   |   |   | 128 |
| kezdő zárójel  | ( | 4  |        |   |   | 1 |   |   |   |   |   |   |   | 64  |
| végzárójel     | ) | 5  |        |   |   |   | 1 |   |   |   |   |   |   | 32  |
| betű           | x | 6  |        |   |   |   |   | 1 |   |   |   |   |   | 16  |
| szám           | 2 | 7  |        |   |   |   |   |   | 1 |   |   |   |   | 8   |
| függvény vége  | , | 8  |        |   |   |   |   |   |   | 1 |   |   |   | 4   |
| egyenlőség     | = | 9  |        |   |   |   |   |   |   |   | 1 |   |   | 2   |
| beolvasás vége | x | 10 |        |   |   |   |   |   |   |   |   | 1 | 1 | 1   |

5. ábra



MF(I)-ben, az egyes Hollerith-karakterek belső reprezentációja található, az 5. ábrán látható sorrendben. /MF(I)-ben elhelyezett elem belső reprezentációja I./

|    |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|    | - | . | + | ( | ) | A | B | C | D | E  | F  | G  | H  | I  | J  | K  | L  | M  | N  | O  | P  | Q  | R  | S  | T  | U  | V  | W  | X  | Y  | Z  |
| MF | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|    | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | ,  | =  | #  | ▽  |
| MF | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 |

6. ábra

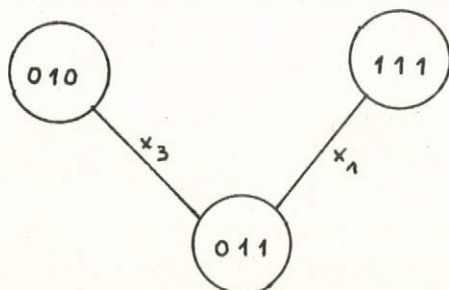


### 3. MINIMALIZÁLÓ ILL. EGYSZERÜSÍTŐ PROGRAMRÉS Z

Röviden összefoglaljuk az algoritmusokban alkalmazott gráf módszer lényegét és leírjuk annak számítógépes realizációját is.

#### Definíciók

- Egy  $f(x_1, x_2, \dots, x_n)$   $n$  változós teljesen meghatározott Boole-függvény értelmezési tartománya az  $n$ -dimenziós bináris tér. E tér pontjainak tehát  $n$  koordinátája van  $\alpha_1, \alpha_2, \dots, \alpha_n$ , ahol  $\alpha_i = 0$  vagy  $1$  lehet.
- Egy teljesen meghatározott  $n$  változós függvény  $f(\alpha_1, \alpha_2, \dots, \alpha_n)$  helyettesítési értéke  $0$  vagy  $1$  lehet. A függvény  $0$ - ill.  $1$ -pontjának nevezzük az  $(\alpha_1, \alpha_2, \dots, \alpha_n)$  pontot, ha  $f(\alpha_1, \alpha_2, \dots, \alpha_n) = 0$  ill.  $1$ .
- Az  $(\alpha_1, \alpha_2, \dots, \alpha_n)$  pont és az  $x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot \dots \cdot x_n^{\alpha_n}$  minterm között kölcsönösen egyértelmű megfeleltetés áll fenn.
- Azt mondjuk, hogy az  $(\alpha_1, \alpha_2, \dots, \alpha_n)$  és  $(\beta_1, \beta_2, \dots, \beta_n)$  pontok távolsága  $k$ , ha pontosan  $k$  koordinátában különböznek.
- Azokat a pontokat, amelyeknek távolsága  $1$ , szomszédos pontoknak nevezzük.
- Egy teljesen meghatározott  $n$  változós függvény gráfja egy olyan gráf, amelynek csúcsai a függvény  $1$ -pontjai, élei pedig a szomszédos  $1$ -pontokat kötik össze /7. ábra/, az éleken feltüntetjük azt a változót, amely szerint a megfelelő két pont szomszédos.



7. ábra

Az  $f = \bar{x}_1 x_2 + x_2 x_3$  gráfja



- Azt mondjuk, hogy egy konjunkció illeszkedik egy  $P$  pontra, ha ez a pont a konjunkciónak  $l$ -pontja.

A gráf módszer a függvény  $l$ -pontjaira illeszkedő konjunkciók között keresi a függvény primimplikánsait.

Annak eldöntésére, hogy egy  $n$  változós függvény egy  $P$   $l$ -pontjára illeszkedő  $k$  változós konjunkció implikáns-e vagy azt kell megállapítani, hogy a konjunkció minden  $l$ -pontja  $/2^{n-k}$  darab/ a függvénynek is  $l$ -pontja-e vagy azt, hogy a konjunkció  $l$ -pontjai között nem szerepel a függvény egyetlen  $0$ -pontja sem.

Ha az  $f$   $n$ -változós függvény egy  $l$ -pontjának  $\ell$  szomszédja van, akkor az  $f$  függvényt implikáló erre a pontra illeszkedő konjunkció változóinak száma  $n-\ell$ -nél kevesebb nem lehet.

Ha az  $f$  függvény valamely  $P$  pontjára a szomszédos pontok száma  $n-k$ , és a  $P$ -re ezen  $n-k$  koordináta szerint illeszkedő konjunkció implikáns, akkor ez lényeges primimplikáns. Ha a  $P$  pontra az  $n-k$  koordináta szerint illeszkedő konjunkció nem implikáns, akkor a  $P$ -re egy-nél több primimplikáns illeszkedik. Ebben az esetben a feladat a  $P$  pontra illeszkedő összes primimplikáns felírása. Ebből a célból a szomszédos pontokhoz vezető  $n-k$  koordináta közül ki kell választani azon  $n-k-i$  ( $i=1,2,\dots,n-k-1$ ) elemű koordináta-csoportokat, amelyek szerint a  $P$ -re illeszkedő konjunkciók primimplikánsok.

A kiválasztás lényegében az  $n-k$  elem  $i$ -ed osztályu kombinációinak előállítását jelenti. Erre olyan eljárást dolgoztunk ki, amely azon kombinációk nagy részét, amelyeknek megfelelő koordináták szerint a  $P$ -re illeszkedő konjunkciót valamely már megtalált primimplikáns elnyelne, nem is állítja elő.

Könnyű belátni, hogy ha az  $\ell-1$ -ed osztályu kombinációkból úgy képezzük az  $\ell$ -ed osztályuakat, hogy mindegyikhez hozzávesszük egyenként a bennük szereplő legnagyobb számnál nagyobb számokat, akkor az összes  $\ell$ -ed osztályu kombinációt megkapjuk és mindegyiket csak egyszer.

Ezt a tényt használjuk fel a  $P$ -re illeszkedő konjunkciók kiválasztásánál, ahol a szomszédszám csökkentését növekvő osztályu kombinációk

alapján végezzük. Vesszünk egy  $C$  kombinációt, és ehhez kiválasztjuk a  $P$ -re illeszkedő azon konjunkciót, amely a  $C$ -ben nem szereplő koordináták szerint illeszkedik  $P$ -re. Ha a  $C'$  a  $C$ -nél magasabb osztályu olyan kombináció, amelyben  $C$  minden eleme szerepel, akkor a  $C'$ -hez kiválasztásra kerülő konjunkciót a  $C$ -hez kiválasztott konjunkció elnyeli. Ebből következik, hogyha a kombinációk képzése során eljutunk egy olyan  $C$  kombinációhoz, amelyhez kiválasztott konjunkció primimplikáns, akkor a  $C$ -vel a fenti relációban álló  $C'$  kombinációkhoz kiválasztott konjunkciókat a  $C$ -hez kiválasztott primimplikáns elnyelné, tehát a  $C'$ -k képzése szükségtelen.

#### Az eljárás a következő:

Legyen a  $P$  pontnak  $S$  szomszédja. Tegyük fel, hogy a szomszédok számát minimálisan  $j$ -vel kell csökkenteni. Jelölje  $ky$  a képzett kombináció osztályát,  $nv$  a képzés során kapott  $v$ -ed osztályu kombinációk számát,  $NI(i)$  a  $v$ -ed osztályu kombinációkat ( $i=1,2,\dots,nv$ ),  $Nl(i)$  az első osztályu kombinációkat ( $i=1,2,\dots,s$ ).

1.  $ky \leftarrow 0$ ,  $nv \leftarrow 1$ ,  $NI(1) \leftarrow 0$ ,  $Nl(i) \leftarrow i$  ( $i=1,2,\dots,s$ )
2.  $nw \leftarrow 0$ ,  $ky \leftarrow ky+1$ ,  $v \leftarrow ky$
3.  $m \leftarrow 0$
4.  $m \leftarrow m+1$
5. Ha  $NI(m) < Nl(v)$ , akkor a 6. lépés, egyébként a 9. lépés következik
6.  $NI(m)$ -ből és  $Nl(v)$ -ből képezzük a  $ky$ -ad osztályu  $C$  kombinációt
7. Ha  $ky < j$ , akkor a 14. lépéssel folytatjuk az eljárást
8.  $nw \leftarrow nw+1$   
 $NI\ l(nw) \leftarrow C$
9. Ha  $m \neq nw$ , akkor a 4. lépés következik
10. Ha  $v=s$ , akkor  $v \leftarrow v+1$  és a 3. lépés következik
11. Ha  $nw=0$ , akkor az eljárás befejeződik, egyébként  $NI(\ell) \leftarrow NI\ l(\ell)$ ,  
 $\ell=1,2,\dots,nw$ ,  $nv \leftarrow nw$
12. Ha  $ky=1$ , akkor  $Nl(k) \leftarrow NI(k)$ ,  $k=1,2,\dots,nw$ ;  $s \leftarrow nw$
13. Ha  $ky=s$ , akkor az eljárás befejeződik, egyébként a 2. lépéssel folytatódik
14. Ha  $C$ -t a további kombinációk képzéséhez meg kell tartani, akkor a 8. lépéssel folytatódik az eljárás
15. A 9. lépés következik.



A primimplikánsokat kereső algoritmusok, a függvény 1-pontjainak ekvivalencia osztályokba sorolását is elvégzik. Ehhez a megtalált primimplikáns minden pontjához hozzárendelünk egy, a primimplikánsra utaló indexet. Így, amikor a primimplikánsok felírása befejeződik, az ekvivalencia osztályok is kialakulnak, ugyanis pontosan azok az 1-pontok tartoznak egy ekvivalencia osztályba, amelyek azonos indexszel rendelkeznek.

#### A függvény gráfjának számítógépes realizációja

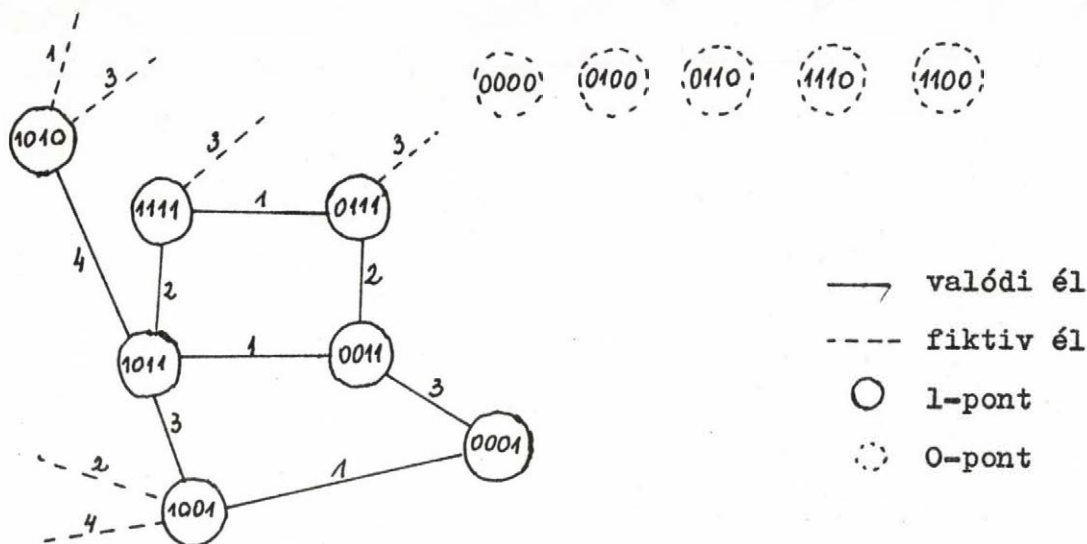
Az  $x_1, x_2, \dots, x_n$  változókhoz az  $x_i \leftrightarrow 2^{i-1}$  megfeleltetést vezetjük be. Az  $(\alpha_1, \alpha_2, \dots, \alpha_n)$  pontnak e szerint az a bináris szám felel meg, amelyben az  $i$ -l-edik helyértéken az  $\alpha_i$  áll.

A függvény gráfjának gépi reprezentációjában a pontokat az előző módon, az éleket pedig pontokhoz rendelve realizáljuk úgy, hogy minden ponthoz egy olyan bináris szám is tartozik, amelyben az  $x_i$ -nek megfelelő  $i$ -l-edik helyértéken 0 áll, ha az illető pontnak az  $x_i$  változó szerint van szomszédja és 1 egyébként /szomszédkonfiguráció/.

A nem teljesen meghatározott Boole-függvények / $\phi$ -Boole-függvények/ vagy függvényrendszerek esetében az előzőekhez hasonló elvek alapján keressük a primimplikánsokat ill. az együttes primimplikánsokat. Az algoritmusok a függvényekhez tartozó megfelelően általánosított gráf alapján határozzák meg az együttes primimplikánsokat és az ekvivalencia osztályokat.

A  $\phi$ -Boole-függvény gráfja tulajdonképpen két gráf: az 1-pontok gráfja és a 0-pontok gráfja. A 0-pontok gráfja a 0-pontoknak megfelelő izolált pontokból áll. Az 1-pontok gráfja egy olyan gráf, amelynek csucsai a függvény 1-pontjai. A gráf éleit két csoportba osztjuk, valódi és fiktív élekre. A valódi élek az  $f$  függvény szomszédos pontjait kötik össze, az egy adott pontra illeszkedő fiktív élek pedig azt mutatják, hogy az adott pontnak van szomszédos pontja  $f$ -ben. A gráf éleihez minőségüktől függetlenül hozzárendeljük annak a változónak megfelelő koordinátát, amelyben a szomszédos pont eltér az adott ponttól /lásd 8. ábra/





Az  $f = x_2 x_3 x_4 \vee \bar{x}_2 x_4 \vee x_1 \bar{x}_2 x_3$ ,  $f_0 = x_2 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$   $\Phi$ -Boole-függvény-tartozó  $\Phi$ -adatgráf

8. ábra

Ez a  $\Phi$ -adatgráf lehetővé teszi a konjunkciók implikáns voltának az előzőeknél egyszerűbb eldöntését. Nevezetesen, implikáns minden olyan konjunkció, amelynek a függvény egyetlen 0-pontja sem pontja.

Az  $F = \{f_1, f_2, \dots, f_M\}$  Boole-függvényrendszer gráfja a komponens függvények gráfjainak egyesítéseként áll elő, úgy, hogy az egyes pontokhoz a komponens függvényekhez tartozást mutató indexet rendelünk. Az egy csucsra illeszkedő élek halmaza a komponens függvények gráfjaiban az adott csucsokra illeszkedő élek halmazainak egyesítéseként áll elő. Az élek közül valódi lesz minden olyan él, amely legalább egy gráfban valódi élként szerepel.

Példaként megadjuk az  $F = \{f_1, f_2, f_3, f_4\}$  függvényrendszer együttes gráfját.

$$f_{1,1} = x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 x_2 x_3 \bar{x}_4$$

$$f_{2,1} = x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 x_2 x_3 \bar{x}_4 \vee x_1 \bar{x}_2 x_3 x_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4$$

$$f_{3,1} = x_1 \bar{x}_2 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 x_4$$

$$f_{4,1} = x_1 \bar{x}_3 \bar{x}_4 \vee x_1 x_2 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 x_4$$

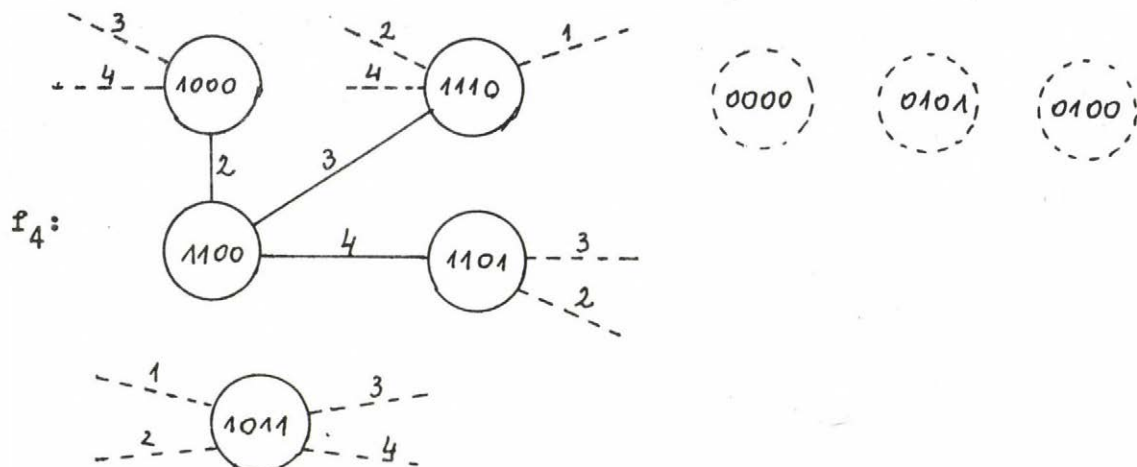
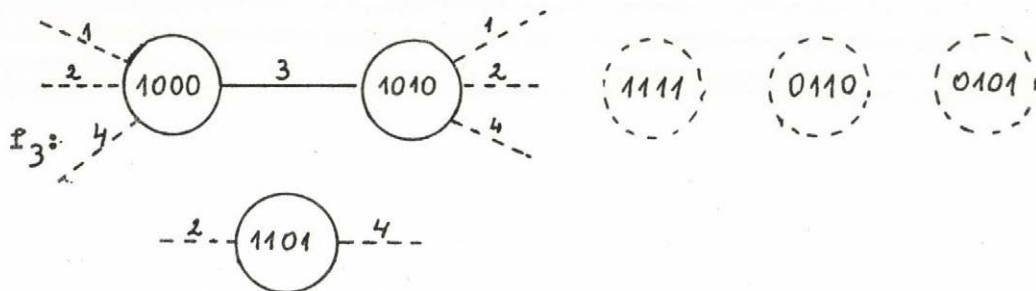
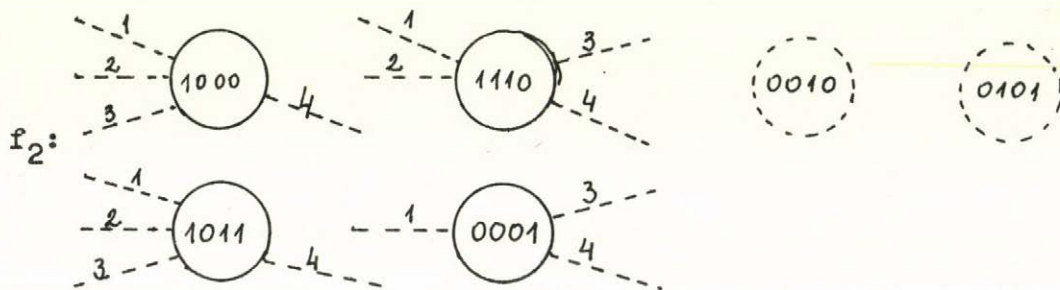
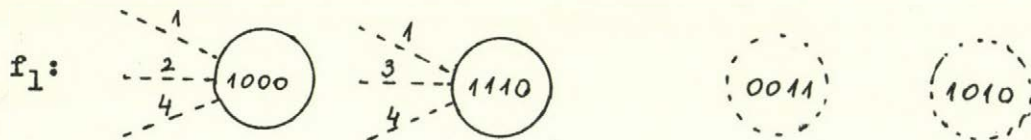
$$f_{1,0} = \bar{x}_1 \bar{x}_2 x_3 x_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4$$

$$f_{2,0} = \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4$$

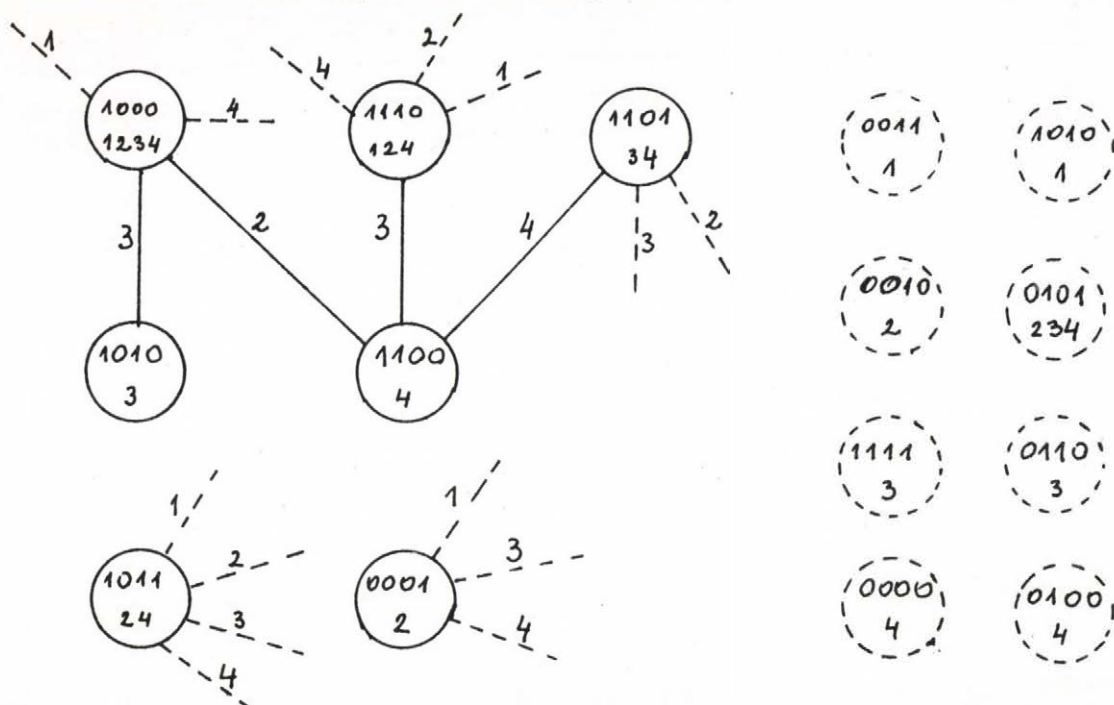
$$f_{3,0} = x_1 x_2 x_3 x_4 \vee \bar{x}_1 x_2 x_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4$$

$$f_{4,0} = \bar{x}_1 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4$$

Az egyes koordinátafüggvények gráfjai:



Az F-hez tartozó együttes Boole-gráf:



9. ábra

A  $\Phi$ -Boole-függvény gráfjainak gépi reprezentációja abban tér el a Boole-függvény gráfjától, hogy a szomszéd konfigurációban a fiktív élek is szerepelnek.

A Boole-függvényrendszer gráfjának gépi reprezentációjában az előzőekhez képest az az eltérés, hogy az egyes gráfpontokhoz két gépi szó tartozik, a pont és a pont indexének reprezentálására.

Az egyszerűsítő és minimalizáló programban lévő algoritmusok.

A primimplikáns keresés mint már említettük a gráf módszerrel történik, amely a függvény /rendszer/ 1-pontjaira illeszkedő konjunkciók közül kiválasztja a primimplikánsokat.

A konjunkciók felírásához a függvény pontjait szomszédszámuk növekvő sorrendje szerint járjuk be. Ez azért előnyös, mert egyrészt a kisebb szomszédszámú pontoknál kevesebb a variálási lehetőség még akkor is, ha nem illeszkedik lényeges primimplikáns erre a pontra, másrészt mivel  $s = \log_2 L$ -nél  $/L$  a függvény pontjainak száma/ több szomszéddal rendelkező pontokra nem létezik primimplikáns - a lényeges



primimplikánsoknak az elsők közötti megtalálása biztosítva van.

A programrendszeren belül két szubrutin létezik arra a célra, hogy előállítsa a függvény ill. függvényrendszer gráfjának gépi reprezentációját és a pontokat szomszédszámuk növekvő sorrendje szerint rendezze. A "RENDEZ" szubrutin egy teljesen meghatározott függvény esetén, a "PR2" szubrutin függvényrendszer esetén végzi el ezt a feladatot.

A primimplikáns kereső algoritmusoknak két fajtáját tartalmazza a programrendszer mind egy teljesen meghatározott függvény, mind függvényrendszer esetére. Az egyik fajta algoritmusok minden primimplikánst felírnak, mivel a minimalizálás előkészítése a céljuk, a másik fajta algoritmusok a primimplikánsok egy olyan halmazát állítják elő, amelynek felhasználásával egyszerűsített, nem redundáns, diszjunktív normálformák előállíthatók. /Az algoritmusokban az, hogy valamely " $P_i$  pont fel nem dolgozandó" azt jelenti, hogy már megtalált primimplikánsok fedik, és újabb primimplikánsok keresésére az algoritmus nem kell, hogy felhasználja./ Az algoritmusok az alábbiak:

1. Egy teljesen meghatározott függvény összes primimplikánsát felíró algoritmus, amelyet a programrendszerben a "MINDPR" szubrutin realizál.

Jelölje  $L$  a függvény pontjainak számát,  $P_i$  az  $i$ -edik pontot,  $s_i$  ill.  $K_i$  az  $i$ -edik pont szomszédainak számát ill. szomszédaihoz vezető koordináták halmazát,  $n$  az  $f$  függvény változóinak számát.

#### 1. rész:

1.  $i \leftarrow 0$
2.  $i \leftarrow i+1$
3. Ha  $s_i > \lceil \log_2 L \rceil$ , akkor áttérés a 2. részre
4.  $P_i$ -re  $K_i$  szerint illeszkedő  $p_i$  konjunkció felírása, azaz  $p_i$ -nek a  $K_i$ -ben nem szereplő koordinátáinak megjelölése
5. Ha  $p_i$  nem implikánsa  $f$ -nek, akkor az algoritmus a 2. lépéssel folytatódik
6.  $p_i$  által lefedett pontokat a  $p_i$ -re utaló indexszel látjuk el. E pontok közül elhagyjuk a kritériumnak eleget tevő, azaz az elhagyható pontokat: megfelelő csucok és élek törlése a gráfból,  $L$  csökkentése.
7. Ha  $L=0$ , akkor az algoritmus az 1. lépéssel folytatódik.

8. Az algoritmus befejeződik, mivel már az első rész megadja az összes primimplikánst.

## 2. rész:

1.  $i \leftarrow 0$
  2.  $i \leftarrow i+1$
  3. Ha  $s_i = n$ , akkor az eljárás befejeződik
  4. Ha az  $1, 2, \dots, n$  számok  $k$ -ad osztályu kombinációit előállító algoritmusunk szerint nem lehet képezni  $C \subset K_1$ -t, akkor az eljárás a 8. lépéssel folytatódik
  5.  $C$  kiválasztása
  6.  $P_i$ -re  $C$  szerint illeszkedő  $p$  konjunkció felírása  
Ha  $p \not\rightarrow f$  vagy  $p$  az előző iterációs lépések során felírásra került, akkor az eljárás a 3. lépéssel folytatódik
  7.  $p$  pontjainak  $p$ -re utaló indexszel való ellátása. Az eljárás a 4. lépéssel folytatódik
  8. A  $P_i$  pontot megjelöljük /a későbbiekben figyelmen kívül hagyható/
  9. Ha  $L \neq i$ , akkor az eljárás a 2. lépéssel folytatódik, egyébként az eljárás befejeződik.
2. Egy teljesen meghatározott függvény egyszerűsítését biztosító primimplikánshalmazt felíró algoritmus. Az algoritmust a programrendszerben a "PRIMI" szubrutin realizálja.
1.  $i \leftarrow 0$   
 $m \leftarrow L$
  2.  $i \leftarrow i+1$
  3. Ha  $P_i$  nem feldolgozandó, akkor áttérés a 2. lépésre
  4. Ha  $s_i = n$ , akkor a 11. lépéssel folytatódik az algoritmus
  5. Felírjuk  $P_i$ -re a  $K_i$  szerint illeszkedő  $p$  konjunkciót
  6. Ha  $p$  nem implikánsa  $f$ -nek, akkor áttérés a 11. lépésre
  7.  $p$  pontjait ellátjuk a  $p$ -re utaló indexszel és a figyelmen kívül hagyhatóság jelével  
 $m$  feldolgozandó pontok száma
  8. Ha  $p$  nem lényeges primimplikáns, akkor áttérés a 11. lépésre
  9. Ha  $m=0$ , akkor az algoritmus befejeződik
  10. Ha  $i=m$ , akkor az algoritmus befejeződik, egyébként a 2. lépéssel folytatódik



11. Ha  $K_i$ -ből már nem lehet primimplikáns előállítására alkalmas  $C$  koordinátahalmazt kiválasztani, akkor az eljárás a 9. lépéssel folytatódik
  12.  $C$  kiválasztása  
 $P_i$ -re  $C$  szerint illeszkedő  $p$  konjunkció felírása. Áttérés a 6. lépésre.
3. Az  $F = \{f_i\} (i=1,2,\dots,M)$  függvényrendszer együttes primimplikánsainak irredundáns normálformák felírását biztosító halmazát előállító algoritmus. Az algoritmust a programrendszerben a "P3" szubrutin realizálja.

Jelölje  $E$  a függvény  $l$ -pontjainak,  $Q$  a  $O$  pontjainak halmazát,  $L$  pedig az  $E$  elemeinek számát.

1.  $i \leftarrow 0$ ;  $m \leftarrow L$
2.  $i \leftarrow i+1$
3. Ha  $P_i$  fel nem dolgozandó, akkor áttérés a 2. lépésre
4. Ha  $s_i = n$ , akkor áttérés a 14. lépésre
5.  $P_i \in E$ -re  $K_i$  szerint illeszkedő  $p$  konjunkció felírása
6. A  $p$  konjunkció  $G$  indexhalmazának kiszámítása
7. Ha  $G$  üres, akkor áttérés a 14. lépésre
8. A  $p$  primimplikáns azon pontjainak indexhalmazaiból, amelyek  $F$ -nek  $l$ -pontjai, töröljük a  $p$  indexeivel közös indexeket, és ellátjuk e pontokat a  $p$ -re utaló indexszel. Az  $F$  azon  $l$ -pontjai, amelyeknek a komponens függvényekhez tartozását mutató indexhalmaza üres, fel nem dolgozandó pontok.
9. Ha  $P_i$ -nek a komponens függvényekhez tartozást mutató indexhalmaza nem üres, akkor áttérés 14. lépésre
10. Ha  $p$  nem lényeges primimplikáns, akkor áttérés a 14. lépésre
11.  $m \leftarrow$  feldolgozandó pontok száma
12. Ha  $m=0$ , akkor az algoritmus befejeződik
13. Ha  $i=L$ , akkor az algoritmus befejeződik, egyébként a 2. lépéssel folytatódik
14. Ha  $K$ -ből már nem lehet primimplikáns előállítására alkalmas  $C$  koordinátahalmazt kiválasztani, akkor az eljárás a 11. lépéssel folytatódik
15.  $C$  kiválasztása  
 $P$ -re  $C$  szerint illeszkedő  $p$  konjunkció felírása  
 Áttérés a 6. lépésre.



4. Függvényrendszerek minimalizálására az összes együttes primimplikáns felírását végző algoritmus az előzőtől annyiban különbözik, hogy az egy pontra illeszkedő primimplikánsok felírása után csak az illető pontot deklaráljuk fel nem dolgozandónak. A programrendszerben ezt az algoritmust a "MINDPT" szubrutin realizálja.

A szubrutinok leírását az alábbiakban adjuk meg.

Az adatok elhelyezése a rutinokban a következő.

Az 1-pontokra vonatkozó információ vagy más szóval egy függvény esetén a függvény gráfjának, függvényrendszer esetén a függvény 1-pontjai általánosított gráfjának gépi reprezentációja az MA(I,J) tömbben (I=6, J=500) található a következőképpen.

Egy teljesen meghatározott függvény esetén:

|         |   |   |
|---------|---|---|
| MA(1,J) | - | 1-pontok belső reprezentációja                        |
| MA(2,J) | - | üres  |
| MA(3,J) | - | az azonos indexű ponthoz tartozó szomszédkonfiguráció |
| MA(4,J) | - | szomszédsszám   |
| MA(5,J) | - | üres  |
| MA(6,J) | - | sorszámozás   |
| MB(I,J) | - | üres  |

Függvényrendszer esetén

|         |   |   |
|---------|---|---|
| MA(1,J) | - | 1-pontok belső reprezentációja  |
| MA(2,J) | - | a megfelelő 1-pont indexe, amely a komponens függvényekhez való tartozást mutatja |
| MA(3,J) | - | szomszédkonfiguráció  |
| MA(4,J) | - | szomszédsszám   |
| MA(5,J) | - | indexmegőrzés   |
| MA(6,J) | - | sorszámozás   |

a 0-pontoknak megfelelő általános gráfot az MB(I,J) (I=2, J=500) tömbben reprezentáljuk

|         |   |   |
|---------|---|---|
| MB(1,J) | - | 0-pontok belső reprezentációja                      |
| MB(2,J) | - | komponens függvényekhez való tartozást mutató index |

### RENDEZ szubrutin

A függvény l-pontjaihoz /mintermjeihez/ hozzárendeli az illető ponttal szomszédos l-pontok számát és egy jelzőszámot, amely megmutatja, hogy a pontnak mely változók szerinti szomszédai l-pontok a függvényben /szomszédkonfiguráció/. Ez a jelzőszám egy olyan n elemű 0-ákból és 1-esekből álló sorozat, amely azon változóknak megfelelő  $2^1$  belső reprezentációk összege, amelyek szerint nincs a pontnak szomszédja a függvény l-pontjai között. Pl. legyen  $x_1\bar{x}_2y$  z a vizsgált pont. A 1110 szomszédkonfiguráció azt mutatja, hogy e pontnak a z változó szerint van szomszédja a függvény l-pontjai között, tehát a lehetséges négy  $\bar{x}_1\bar{x}_2y$  z,  $x_1x_2y$  z,  $x_1\bar{x}_2\bar{y}$  z,  $x_1\bar{x}_2y\bar{z}$  szomszédos pont közül csak az  $x_1\bar{x}_2y\bar{z}$  l-pontja a függvénynek.

A függvény l-pontjait szomszédaik számának növekvő sorrendje szerint rendezi.

Paraméterátadás közös területen történik.

#### Közös terület

|                 |   |   |
|-----------------|---|---|
| /BOOLE/ MA(6,I) | - | MA(1,I) -ben a mintermek listáját átveszi a beolvasó programtól |
|                 |   | MA(3,I) -be a szomszédkonfigurációt                             |
|                 |   | MA(4,I) -be a szomszédsszámot dolgozza ki                       |
| IV              | - | változók száma  |
| IPE             | - | l-pontok száma  |

### PR2 szubrutin

A szubrutin a komponens függvények  $f_{11}$ -jének mintermjeit egy közös listába rendezi. A listában minden minterm egyszer szerepel, és egy index jelzi, hogy az illető minterm mely  $f_{11}$ -nek mintermje. Ugyanígy rendezi listába az  $f_{10}$ -ak mintermjeit is. Kialakítja a komponens függvények l-pontjai /mintermjei/ szomszédkonfigurációit /lásd RENDEZ szubrutin leírása/ és szomszédsszámát. Egy p l-pontnak egy vele szomszédos q pont szomszédja az adott függvényben, ha q nem 0-pont a függvény legalább egy olyan komponens függvényében, amelyben p l-pont. Végül a szubrutin az l-pontokat szomszédsszámuk növekvő sorrendje szerint rendezi.



Paraméterátadás a közös területen történik.

#### Közös terület

/BOOLE/ MA(6,I), MB(2,I)- MA(1,I) ill. MA(2,I)-ben /MB(1,I) ill. MB(2,I)-ben/ kapja az egyes komponens függvények 1-pontjait ill. a komponens függvényre utaló indexet /az egyes komponens függvények 0-pontjait ill. a komponens függvényre utaló indexet/. Ugyanitt alakítja ki a minterm listákat ill. az egyes mintermek indexeit.

MA(3,I)-ben az I-edik 1-pont a szomszédkonfigurációja, MA(4,I)-ben a szomszédszáma található

IV - változók száma

LF - komponens függvények száma

#### PRIMI szubrutin

A függvény 1-pontjaira a pont szomszédkonfigurációja felhasználásával felírja az illető pontot lefedő primimplikánsokat, megjelöli a lényeges primimplikánsokat, és az egyes pontokhoz rendel egy olyan jelzőszámot, amely megmutatja, hogy az illető pontot mely primimplikánsok fedik.

A soron következő pontra felírt primimplikánsok által fedett összes pontot fel nem dolgozandónak deklarálja. Ezzel nem írja fel az összes primimplikánst, csak egy, az egyszerűsítéshez elegendő primimplikánshalmazt.

Paraméterátadás a közös területen történik.

#### Közös terület

/BOOLE/ MA(6,I) MA(1,I), MA(3,I), MA(4,I)-ben átveszi a  
REND szubrutin eredményét

MA(6,I)= I a kiinduláskor, a program futása során pedig  
a még feldolgozandó pontok címei találhatók  
benne

MC(5,100) MC(1,I) I-edik primimplikáns 1. komponense  
MC(2,I) I-edik primimplikáns 2. komponense  
MC(3,I) I-edik primimplikáns  $2^k$  alaku belső



$MC(4, I) = I/23$   
 $ID(6, 500)$  reprezentációja, ahol  $I \equiv k \pmod{23}$  és  $k < 23$   
 $ID(J, I)$ -ben az  $I$ -edik pontot lefedő primimplikánsok  $2^k$  alakú belső reprezentációinak összege szerepel.

### Munkatömbök

$NI(23)$ ,  $NI(300)$ ,  $NI(300)$  -  $n$  elem  $k$ -ad osztályu kombinációi képzésénél felhasznált tömbök

|           |   |                  |             |
|-----------|---|------------------|-------------|
| $NI(23)$  | - | $\binom{n}{1}$   | kombinációk |
| $NI(300)$ | - | $\binom{n}{k}$   | kombinációk |
| $NI(300)$ | - | $\binom{n}{k+1}$ | kombinációk |

### P3 szubrutin

A szubrutin a minterm listában szereplő 1-pontokra felírja az illető pontot tartalmazó együttes primimplikánsokat, és az egyes primimplikánsokhoz indexet rendel, amely a primimplikáns által implikált függvények  $2^i$  alakú belső reprezentációinak összege. Az egyes pontokhoz olyan jelzőszámot rendel, amely megmutatja, hogy az illető pontot mely primimplikánsok fedik. Az 1-pontok indexében  $(MA(2, I))$  csak azon függvényekhez való tartozást hagyja meg, amelyeket nem fedik le még a kapott primimplikánsok. Az 0 indexű pontok lefedettek, tehát fel nem dolgozandók. Ily módon természetesen nem kapjuk meg az összes primimplikánst, csak egy, az egyszerűsítéshez elegendő együttes primimplikánshalmazt.

### Közös terület

/BOOLE/  $MA(6, 500)$ ,  $MB(2, 500) - MA(1, I)$ ,  $MA(2, I)$ ,  $MA(3, I)$ ,  $MA(4, I)$ ,  
 $MB(1, I)$ ,  $MB(2, I)$  tartalmát PR2 szubrutinban határozzuk meg.  $MA(5, I)$  munkatömbként szerepel a  $MA(2, I)$ -beli indexek változásának regisztrálására.  $MA(6, I)$  a szubrutin elején  $MA(6, I) = I$  értéket kap, a program futása során a még feldolgozandó pontok  $MA(1, 500)$ -beli címeit tartalmazza.

$MC(5,100)$        $MC(1,I)$  I-edik primimplikáns 1. komponense  
 $MC(2,I)$  I-edik primimplikáns 2. komponense  
 $MC(3,I)$  I-edik primimplikáns  $2^k$  alakú belső  
 reprezentációja, ahol  $I \equiv k \pmod{23}$  és  
 $k < 23$   
 $MC(4,I) = I/23$   
 $MC(5,I)$  I-edik primimplikáns által lefedett függ-  
 vények  $2^k$  alakú reprezentációinak össze-  
 ge szerepel.

### Munkatömbök

$NI(23)$ ,  $NI(300)$ ,  $NI(300)$  - n elem k-ad osztályu kombinációi kép-  
 zésénél felhasznált tömbök

$NI(23) \quad \binom{n}{1}$  - kombinációk  
 $NI(300) \quad \binom{n}{k}$  - kombinációk  
 $NI(300) \quad \binom{n}{k+1}$  - kombinációk

### MINDPR szubrutin

A függvény l-pontjaira a pont szomszédkonfigurációja felhasználásával  
felírja az illető pontot lefedő primimplikánsokat, megjelöli a lénye-  
ges primimplikánsokat, és az egyes pontokhoz rendel egy olyan jelző-  
 számot, amely megmutatja, hogy az illető pontot mely primimplikánsok  
 fedik.

Fel nem dolgozandónak csak az éppen feldolgozott pontot tekintjük,  
 így biztosítjuk az összes primimplikáns felírását.

A szubrutinra vonatkozó egyéb információ azonos a PRIMI szubrutin-  
 beliekkel.

### MINDPT szubrutin

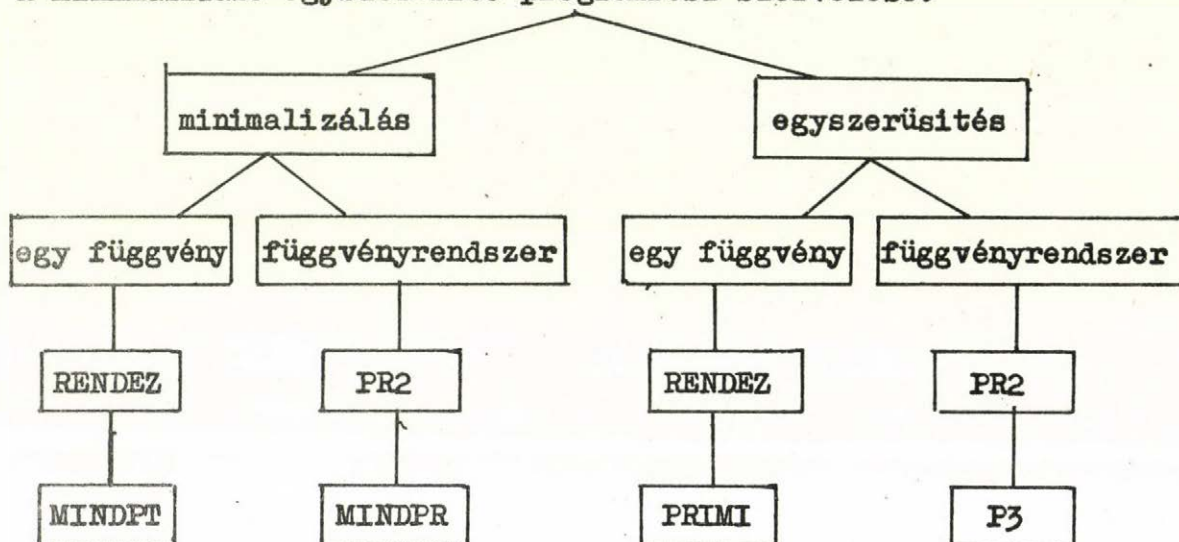
A szubrutin a minterm listában szereplő l-pontokra felírja az illető  
 pontot tartalmazó együttes primimplikánsokat, és az egyes primimpli-  
 kánsokhoz indexet rendel, amely a primimplikáns által implikált függ-  
 vények  $2^1$  alakú belső reprezentációinak összege. Az egyes pontokhoz

olyan jelzőszámot rendel, amely megmutatja, hogy az illető pontot mely primimplikánsok fedik.

Az éppen feldolgozott pontot fel nem dolgozandónak tekinti. Ezzel az összes együttes primimplikáns felírását adja.

A szubrutinra vonatkozó egyéb információ azonos a P3 szubrutinbeliével.

A minimalizáló-egyszerűsítő programrész szervezése:



10. ábra



#### 4. KIIRO ILL. KIÉRTÉKELŐ PROGRAMRÉSZE

Ez a program lényegében a KIIRO szubrutinból áll.

A program a primimplikáns lista, minterm-lista és az index-listák alapján egy függvény és függvényrendszer esetén egységesen végzi az eredményadatok kiértékelését.

Megvizsgálja az egyes pontokra vonatkozóan, hogy az illető pontot mely primimplikánsok fedik, és az IE nevű tömbbe elhelyezi az éppen feldolgozott pontot fedő primimplikánsokat. Megállapítja, hogy a pontot fedő primimplikánsok között van-e lényeges primimplikáns. Az MQ tömbben az MQ(I)-ben 0 áll, ha az I-edik primimplikáns nem lényeges primimplikáns és azokat a függvényeket mutató index, amelyekre nézve az I-edik primimplikáns lényeges primimplikáns. A lényeges primimplikánsok által fedett pontok indexeit a lefedésnek megfelelően módosítja. Összegyűjti az ekvivalens, még nem teljesen lefedett pontokat, és előállítja a diszjunkt halmazok listáját. Elkészíti a lefedési táblázatot a még le nem fedett diszjunkt halmazokra, és a nem lényeges, valamint a nem minden indexük szerint lényeges primimplikánsokra. Majd kinyomtatja a primimplikáns listát.

A KIIRO szubrutin az alábbi táblázatokat írja ki:

a/ A feldolgozott függvények és a hozzájuk tartozó  $2^i$  alakú indexek listájának, és a függvények és a változók számának kiírása az alábbi formában:

| INDEX             | FV. NÉV |
|-------------------|---------|
| 1                 | FK1     |
| 2                 | G5      |
| 32                | Y       |
| FÜGGVÉNYEK SZÁMA: | 3       |
| VÁLTOZÓK SZÁMA:   | 7       |

b/ A Boole-függvény vagy függvények összes primimplikánsainak és lényeges primimplikánsainak listáját. Több függvény együttes egyszerűsítése esetében ez a lista egy mutatószám listával bővül, amely jelzi, hogy az illető primimplikáns mely függvényeknek primimplikánsa ill. lényeges primimplikánsa.

A lista formája:

|    | IND         | L.PR.IMPL. | IND.  | PR.IMPL.  |
|----|-------------|------------|-------|-----------|
| 1. | $n_1 \ k_1$ | a. -b.c    | -     | -         |
| 2. | ...         |            | $n_2$ | -a.c.d.-b |
| 3. | ...         |            |       |           |
| ⋮  |             |            |       |           |

$n_1$  azon függvények indexeinek összege, amelyeknek az illető primimplikáns implikánsa.

$k_1$  azon függvények indexeinek összege, amelyeknek az illető primimplikáns lényeges primimplikánsa.

- c/ A lényeges primimplikánsok által le nem fedett függvényt pontok diszjunkt halmazainak listája, és egy index lista, amely egyrészt azt mutatja, hogy az egyes diszjunkt halmazok mely függvényeknek pontjai F.IND. másrészt pedig azt, hogy a lényeges primimplikánsok mely függvényekben nem fedik le a diszjunkt halmazt F.MUNK.I. .

| SORSZ.     | F.IND. | F.MUNK.I. | DT.E.SORSZ. | DT.ELEMEI |
|------------|--------|-----------|-------------|-----------|
| 1.         | $j_1$  | $k_1$     | $i_1$       | $m_1$     |
|            | ⋮      | ⋮         | $i_2$       | $m_2$     |
|            | ⋮      | ⋮         | ⋮           | ⋮         |
|            | ⋮      | ⋮         | ⋮           | ⋮         |
|            | ⋮      | ⋮         | ⋮           | ⋮         |
|            | ⋮      | ⋮         | $i_k$       | $m_k$     |
| 2. . . . . |        |           |             |           |
| ⋮          |        |           |             |           |
| ⋮          |        |           |             |           |

- d/ Lefedési táblázat, amely megmutatja, hogy az egyes diszjunkt halmazokat mely primimplikánsok fedik le.

A táblázat:

| PONT.    | SORSZ. | $i_1$ | $i_2$ | ... |
|----------|--------|-------|-------|-----|
| PRI.     | IND.   | $j_1$ | $j_2$ |     |
| $k_1$    | $n_1$  | 1     | 1     |     |
| $k_2$    | $n_2$  | 0     | 1     |     |
| $\vdots$ |        |       |       |     |

ahol  $i_s$  a diszjunkt halmaz első elemének sorszáma,  $j_i$  a diszjunkt halmaz indexe,  $k_j$  a primimplikáns sorszáma a listában,  $n_j$  a primimplikáns indexe.

Megjegyzés: Az 1. mellékletben néhány függvényrendszer egyszerűsítési és minimalizálási eredménye található.

Paraméterátadás a közös területen történik.

#### Közös terület

MA(6,300) - adatátvétel  
 ID(6,300) - adatátvétel  
 JV(100) - változólista adatátvétel

#### Megjegyzés:

A programrendszer az adatként szereplő függvények megfelelő megadásával alkalmas

- teljesen meghatározott Boole-függvény negálására

Módja: keressük az  $f$  függvény negáltját. Adatként meg kell adni

-  $f$  függvénynévvel  $f$  diszjunktív normálformáját és valamely  $g$  függvénynévvel  $f$  változóinak egy az  $f$ -et implikáló konjunkcióját. Pl.  $f=ab+acx$  negáltját keressük. Adatok:

-  $f=a.b+\neg a.c.x$ ,  $g=a.b.c.x$   $\neq$

-  $\phi$ -Boole-függvényt meghatározó bármely két komponens esetén a har-



madik komponens meghatározása.

1.  $f_1, f_\phi$ -ből  $f_0$  előállítása. Adatmegadás:

$$-f_1 = \dots, -f_\phi = \dots \quad \text{☒}$$

2.  $f_0, f_\phi$ -ből  $f_1$  előállítása. Adatmegadás:

$$-f_0 = \dots, -f_\phi = \dots \quad \text{☒}$$

3.  $f_1, f_0$ -ból  $f_\phi$  meghatározása. Adatmegadás:

$$-f_1 = \dots, -f_0 = \dots \quad \text{x} \quad /f_1, f_0 \text{ azonosítója különböző kell, hogy legyen/.}$$

Ezek a lehetőségek biztosítják, hogy a programrendszer működéséhez szükséges  $f_1, f_0$  megadása ne ütközzön nehézségbe.

A 2. mellékletben megadjuk a programrendszer felhasználói leírását.

## IRODALOMJEGYZÉK

- 1 Pásztorné Varga Katalin: Módszerek Boole-függvények minimális vagy nem redundáns,  $\{\wedge, \vee, \neg\}$  vagy  $\{\text{NOR}\}$  vagy  $\{\text{NAND}\}$  bázisbeli zárójeles vagy zárójel nélküli formuláinak előállítására, MTA SzTAKI Tanulmányok 1/1973.
- 2 B. Randell - I. Russel: ALGOL 60 implementation, Acad. Press. 1964.





## MELLÉKLETEK



ILLUSZTRÁCIÓ NÉHÁNY FÜGGVÉNYRENDSZERNEK A PROGRAMRENDSZER  
ÁLTAL VALÓ EGYSZERÜSÍTÉSÉRE ÉS MINIMALIZÁLÁSÁRA

1.1 Egy teljesen meghatározott, 26 db 1-pontot tartalmazó ötváltozós függvény egyszerűsítése.

Felhasznált gépidő 40 mp.

$F_{26} = -X_1 \cdot -X_2 \cdot -X_3 \cdot -X_4 \cdot -X_5 + X_1 \cdot -X_2 \cdot -X_3 \cdot -X_4 \cdot -X_5 + -X_1 \cdot X_2 \cdot -X_3 \cdot -X_4 \cdot -X_5 + -X_1 \cdot -X_2 \cdot X_3$   
 $\cdot -X_4 \cdot -X_5 + -X_1 \cdot -X_2 \cdot -X_3 \cdot X_4 \cdot -X_5 + -X_1 \cdot -X_2 \cdot -X_3 \cdot -X_4 \cdot X_5 + X_1 \cdot -X_2 \cdot X_3 \cdot X_4 \cdot -X_5 + X_1 \cdot -X_2 \cdot$   
 $X_3 \cdot -X_4 \cdot -X_5 + -X_1 \cdot X_2 \cdot X_3 \cdot -X_4 \cdot -X_5 + X_1 \cdot -X_2 \cdot -X_3 \cdot X_4 \cdot -X_5 + -X_1 \cdot X_2 \cdot -X_3 \cdot -X_4 \cdot X_5 + -X_1 \cdot -X_2$   
 $\cdot X_3 \cdot -X_4 \cdot X_5 + -X_1 \cdot -X_2 \cdot -X_3 \cdot X_4 \cdot X_5 + X_1 \cdot X_2 \cdot X_3 \cdot -X_4 \cdot -X_5 + X_1 \cdot X_2 \cdot -X_3 \cdot X_4 \cdot -X_5 + -X_1 \cdot X_2 \cdot X_3$   
 $\cdot X_4 \cdot -X_5 + X_1 \cdot X_2 \cdot -X_3 \cdot -X_4 \cdot X_5 + X_1 \cdot -X_2 \cdot X_3 \cdot -X_4 \cdot X_5 + X_1 \cdot -X_2 \cdot -X_3 \cdot X_4 \cdot X_5 + -X_1 \cdot X_2 \cdot -X_3 \cdot X_4$   
 $\cdot X_5 + -X_1 \cdot -X_2 \cdot X_3 \cdot X_4 \cdot X_5 + X_1 \cdot X_2 \cdot X_3 \cdot X_4 \cdot -X_5 + X_1 \cdot X_2 \cdot X_3 \cdot -X_4 \cdot X_5 + X_1 \cdot X_2 \cdot -X_3 \cdot X_4 \cdot X_5 + X_1 \cdot$   
 $-X_2 \cdot X_3 \cdot X_4 \cdot X_5 + -X_1 \cdot X_2 \cdot X_3 \cdot X_4 \cdot X_5$

\*

A feldolgozandó függvény  
beolvasott alakja:



A feldolgozás eredménye:

## Diszjunkt halmazok listája

| SORSZ. | P IND. | P MUNK.I. | DT.E.SRSZ. | DT.ELEMET           |
|--------|--------|-----------|------------|---------------------|
| 1.     | 1      | 1         | 1          | X1.-X2.-X3.-X4.-X5  |
| 2.     | 1      | 1         | 2          | -X1.X2.-X3.-X4.-X5  |
| 3.     | 1      | 1         | 3          | -X1.-X2.-X3.X4.-X5  |
| 4.     | 1      | 1         | 4          | X1.X2.-X3.X4.-X5    |
| 5.     | 1      | 1         | 5          | -X1.X2.X3.X4.-X5    |
| 6.     | 1      | 1         | 6          | X1.X2.-X3.-X4.X5    |
| 7.     | 1      | 1         | 7          | X1.X2.X3.-X4.X5     |
| 8.     | 1      | 1         | 8          | -X1.X2.X3.X4.X5     |
| 9.     | 1      | 1         | 9          | -X1.-X2.X3.-X4.-X5  |
| 10.    | 1      | 1         | 10         | -X1.-X2.-X3.-X4.X5  |
| 11.    | 1      | 1         | 11         | X1.-X2.X3.X4.-X5    |
| 12.    | 1      | 1         | 12         | -X1.X2.X3.-X4.-X5   |
| 13.    | 1      | 1         | 13         | -X1.X2.-X3.-X4.X5   |
| 14.    | 1      | 1         | 14         | -X1.-X2.X3.-X4.X5   |
| 15.    | 1      | 1         | 15         | X1.X2.X3.-X4.-X5    |
| 16.    | 1      | 1         | 16         | X1.-X2.X3.-X4.X5    |
| 17.    | 1      | 1         | 17         | X1.-X2.-X3.X4.X5    |
| 18.    | 1      | 1         | 18         | -X1.X2.-X3.X4.X5    |
| 19.    | 1      | 1         | 19         | -X1.-X2.X3.X4.X5    |
| 20.    | 1      | 1         | 20         | X1.X2.X3.X4.-X5     |
| 21.    | 1      | 1         | 21         | X1.X2.-X3.X4.X5     |
| 22.    | 1      | 1         | 22         | X1.-X2.X3.X4.X5     |
| 23.    | 1      | 1         | 23         | -X1.-X2.-X3.-X4.-X5 |
| 24.    | 1      | 1         | 24         | X1.-X2.X3.-X4.-X5   |
| 25.    | 1      | 1         | 25         | X1.-X2.-X3.X4.-X5   |
| 26.    | 1      | 1         | 26         | -X1.-X2.-X3.X4.X5   |

Primimplikáns lista:

IND. INDL. L.PRIMIMPL.

|     | IND. | PRIMIMPL.    |
|-----|------|--------------|
| 1.  | 1    | X1.-X2.-X5   |
| 2.  | 1    | -X2.-X3.-X5  |
| 3.  | 1    | -X2.-X4.-X5  |
| 4.  | 1    | -X1.-X3.-X4  |
| 5.  | 1    | -X1.-X4.-X5  |
| 6.  | 1    | X1.-X3.X4    |
| 7.  | 1    | X1.X4.-X5    |
| 8.  | 1    | X2.X3.-X5    |
| 9.  | 1    | -X1.X2.X3.X4 |
| 10. | 1    | X2.-X3.X5    |
| 11. | 1    | X1.X2.-X4.X5 |
| 12. | 1    | -X1.-X2.-X4  |
| 13. | 1    | -X2.X3.-X4   |
| 14. | 1    | -X1.-X2.X5   |
| 15. | 1    | -X2.X3.X5    |

A függvény adatai:

FUGGVENYEK SZAMA: 1

VALTOZOK SZAMA: 5

1-PONTOK SZAMA: 26

## Lefedési táblázat

| PONT | SORSZ | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   | 11   | 12   | 13   |
|------|-------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| PRI  | IND   | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ |
| 1.   | / 1/  | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    | 0    | 0    |
| 2.   | / 1/  | 1    | 0    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 3.   | / 1/  | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 4.   | / 1/  | 1    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    | 0    | 0    | 1    |
| 5.   | / 1/  | 1    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    | 0    | 0    | 0    |
| 6.   | / 1/  | 1    | 0    | 0    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 7.   | / 1/  | 1    | 0    | 0    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 8.   | / 1/  | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    | 0    |
| 9.   | / 1/  | 1    | 0    | 0    | 0    | 0    | 1    | 0    | 0    | 0    | 1    | 0    | 0    | 0    |
| 10.  | / 1/  | 1    | 0    | 0    | 0    | 0    | 0    | 1    | 0    | 0    | 0    | 0    | 0    | 1    |
| 11.  | / 1/  | 1    | 0    | 0    | 0    | 0    | 0    | 1    | 1    | 0    | 0    | 0    | 0    | 0    |
| 12.  | / 1/  | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    | 1    | 0    | 0    | 0    |
| 13.  | / 1/  | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    | 0    | 0    | 0    |
| 14.  | / 1/  | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    | 0    | 0    | 0    |
| 15.  | / 1/  | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| PONT | SORSZ | 14   | 15   | 16   | 17   | 18   | 19   | 20   | 21   | 22   | 23   | 24   | 25   | 26   |
| PRI  | IND   | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ | / 1/ |
| 1.   | / 1/  | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    | 1    | 0    |
| 2.   | / 1/  | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    | 1    | 1    | 0    |
| 3.   | / 1/  | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    | 1    | 0    | 0    |
| 4.   | / 1/  | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    | 0    | 0    | 0    |
| 5.   | / 1/  | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    | 0    | 0    | 0    |
| 6.   | / 1/  | 0    | 0    | 0    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    | 0    |
| 7.   | / 1/  | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    | 0    |
| 8.   | / 1/  | 0    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 9.   | / 1/  | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 10.  | / 1/  | 0    | 0    | 0    | 0    | 0    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 11.  | / 1/  | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 12.  | / 1/  | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    | 0    | 0    | 0    |
| 13.  | / 1/  | 1    | 0    | 1    | 0    | 0    | 1    | 0    | 0    | 0    | 0    | 1    | 0    | 0    |
| 14.  | / 1/  | 1    | 0    | 0    | 0    | 0    | 0    | 1    | 0    | 0    | 0    | 0    | 0    | 1    |
| 15.  | / 1/  | 1    | 0    | 1    | 0    | 0    | 0    | 1    | 0    | 1    | 0    | 0    | 0    | 0    |



1.2 Egy teljesen meghatározott, 26 db. 1-pontot tartalmazó ötváltozós függvény minimalizálása.

Felhasznált gépidő 1 p. 40 mp.

F26=-x1.-x2.-x3.-x4.-x5+x1.-x2.-x3.-x4.-x5+-x1.x2.-x3.-x4.-x5+-x1.-x2.x3  
.-x4.-x5+-x1.-x2.-x3.x4.-x5+-x1.-x2.-x3.-x4.x5+x1.-x2.x3.x4.-x5+x1.-x2.  
x3.-x4.-x5+-x1.x2.x3.-x4.-x5+x1.-x2.-x3.x4.-x5+-x1.x2.-x3.-x4.x5+-x1.-x2  
.x3.-x4.x5+-x1.-x2.-x3.x4.x5+x1.x2.x3.-x4.-x5+x1.x2.-x3.x4.-x5+-x1.x2.x3  
.x4.-x5+x1.x2.-x3.-x4.x5+x1.-x2.x3.-x4.x5+x1.-x2.-x3.x4.x5+-x1.x2.-x3.x4  
.x5+-x1.-x2.x3.x4.x5+x1.x2.x3.x4.-x5+x1.x2.x3.-x4.x5+x1.x2.-x3.x4.x5+x1.  
-x2.x3.x4.x5+-x1.x2.x3.x4.x5

\*

A feldolgozandó függvény  
beolvasott alakja:

A feldolgozás eredménye:

Diszjunkt halmazok listája

| SORSZ. | P IND. | P MUNK.I. | DT.E.SRSZ. | DT.E.ÉMEI           |
|--------|--------|-----------|------------|---------------------|
| 1.     | 1      | 1         | 1          | X1.-X2.-X3.-X4.-X5  |
| 2.     | 1      | 1         | 2          | -X1.X2.-X3.-X4.-X5  |
| 3.     | 1      | 1         | 3          | -X1.-X2.-X3.X4.-X5  |
| 4.     | 1      | 1         | 4          | X1.X2.-X3.X4.-X5    |
| 5.     | 1      | 1         | 5          | -X1.X2.X3.X4.-X5    |
| 6.     | 1      | 1         | 6          | X1.X2.-X3.-X4.X5    |
| 7.     | 1      | 1         | 7          | X1.X2.X3.-X4.X5     |
| 8.     | 1      | 1         | 8          | -X1.X2.X3.X4.X5     |
| 9.     | 1      | 1         | 9          | -X1.-X2.X3.-X4.-X5  |
| 10.    | 1      | 1         | 10         | -X1.-X2.-X3.-X4.X5  |
| 11.    | 1      | 1         | 11         | X1.-X2.X3.X4.-X5    |
| 12.    | 1      | 1         | 12         | -X1.X2.X3.-X4.-X5   |
| 13.    | 1      | 1         | 13         | -X1.X2.-X3.-X4.X5   |
| 14.    | 1      | 1         | 14         | -X1.-X2.X3.-X4.X5   |
| 15.    | 1      | 1         | 15         | X1.X2.X3.-X4.-X5    |
| 16.    | 1      | 1         | 16         | X1.-X2.X3.-X4.X5    |
| 17.    | 1      | 1         | 17         | X1.-X2.-X3.X4.X5    |
| 18.    | 1      | 1         | 18         | -X1.X2.-X3.X4.X5    |
| 19.    | 1      | 1         | 19         | -X1.-X2.X3.X4.X5    |
| 20.    | 1      | 1         | 20         | X1.X2.X3.X4.-X5     |
| 21.    | 1      | 1         | 21         | X1.X2.-X3.X4.X5     |
| 22.    | 1      | 1         | 22         | X1.-X2.X3.X4.X5     |
| 23.    | 1      | 1         | 23         | -X1.-X2.-X3.-X4.-X5 |
| 24.    | 1      | 1         | 24         | X1.-X2.X3.-X4.-X5   |
| 25.    | 1      | 1         | 25         | X1.-X2.-X3.X4.-X5   |
| 26.    | 1      | 1         | 26         | -X1.-X2.-X3.X4.X5   |

Primimplikáns lista:

| IND. | INDL. | L.PRIMIMPL.  | IND. | PRIMIMPL.    |
|------|-------|--------------|------|--------------|
|      |       | Ellenőrizte: | 1    | X1.-X2.-X5   |
| 1,   |       |              | 1    | -X2.-X3.-X5  |
| 2,   |       |              | 1    | -X2.-X4.-X5  |
| 3,   |       |              | 1    | -X1.-X3.-X4  |
| 4,   |       |              | 1    | -X1.-X4.-X5  |
| 5,   |       |              | 1    | -X1.-X2.-X3  |
| 6,   |       |              | 1    | -X2.-X3.X4   |
| 7,   |       |              | 1    | X1.-X3.X4    |
| 8,   |       |              | 1    | X1.X4.-X5    |
| 9,   |       |              | 1    | X2.X3.-X5    |
| 10,  |       |              | 1    | -X1.X2.X3.X4 |
| 11,  |       |              | 1    | X2.-X3.X5    |
| 12,  |       |              | 1    | X1.X2.-X4.X5 |
| 13,  |       |              | 1    | X1.X3.-X4    |
| 14,  |       |              | 1    | -X1.X4.X5    |
| 15,  |       |              | 1    | -X1.-X2.-X4  |
| 16,  |       |              | 1    | -X2.X3.-X4   |
| 17,  |       |              | 1    | X3.-X4.-X5   |
| 18,  |       |              | 1    | -X1.-X2.X5   |
| 19,  |       |              | 1    | -X1.-X3.X5   |
| 20,  |       |              | 1    | X1.-X2.X3    |
| 21,  |       |              | 1    | X1.-X2.X4    |
| 22,  |       |              | 1    | -X2.X3.X5    |
| 23,  |       |              | 1    | X1.X3.-X5    |
| 24,  |       |              | 1    | -X2.X4.X5    |
| 25,  |       |              | 1    | -X3.X4.X5    |
| 26,  |       |              | 1    |              |

A függvény adatai:

|                   |    |
|-------------------|----|
| FÜGGVÉNYEK SZÁMA: | 1  |
| VALTOZOK SZÁMA:   | 5  |
| 1-PONTOK SZÁMA:   | 26 |



| POINT | SOURCE | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
|-------|--------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| POINT | SOURCE | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| POINT | SOURCE | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| POINT | SOURCE | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |

1.3 Két nem teljesen meghatározott, 6 db 1-pontot tartalmazó, négy-  
változós függvényből álló függvényrendszer egyszerűsítése.

Felhasznált gépidő 16 mp.

```

F1=-U.-X.-Y.-Z+-U.X.Y.Z+U.-X.-Y.-Z+U.X.Y.Z,
-F1=-U.-X.Y.-Z+-U.-X.Y.Z+-U.X.-Y.Z+U.-X.-Y.Z+U.-X.Y.-Z+U.-X.Y.Z+U.X.-Y.-
Z+U.X.-Y.Z,
F2=-U.-X.-Y.Z+-U.-X.Y.Z+-U.X.Y.Z+U.-X.-Y.-Z,
-F2=-U.-X.Y.-Z+-U.X.-Y.Z+U.X.Y.-Z+U.-X.-Y.Z+U.-X.Y.-Z+U.X.-Y.-Z
+U.X.Y.-Z*

```

A feldolgozandó függvény  
beolvasott alakja:

# A feldolgozás eredménye:

## Diszjunkt halmazok listája

| SORSZ. | P IND. | P MUNK.I. | DT.E.SRSZ. | DT.ELEMEI  |
|--------|--------|-----------|------------|------------|
| 1.     | 4      | 4         | 3          | -U,-X,-Y,Z |
| 2.     | 5      | 4         | 5          | -U,X,Y,Z   |
| 3.     | 4      | 4         | 6          | -U,-X,Y,Z  |

## A függvény adatai:

| INDEX | FV.NEV | FUGGVENYEK SZAMA: | 2 |
|-------|--------|-------------------|---|
| 1     | F1     | VALTOZOK SZAMA:   | 4 |
| 1     | F1     | 1-PONTOK SZAMA:   | 6 |
| 4     | F2     |                   |   |
| 4     | F2     |                   |   |

## Lefedési táblázat:

|       |      |       |    |    |   |   |    |   |   |
|-------|------|-------|----|----|---|---|----|---|---|
| ***** |      |       |    |    |   |   |    |   |   |
| *     |      | *     |    | *  |   | * |    | * |   |
| *     | PONT | SORSZ |    | 3  | * | 5 | *  | 6 | * |
| *     |      | *     |    | *  |   | * |    | * |   |
| ***** |      |       |    |    |   |   |    |   |   |
| *     |      | *     |    | *  |   | * |    | * |   |
| *     | PRI  | IND   | /  | 4/ | * | / | 5/ | * | / |
| *     |      | *     |    | *  |   | * |    | * |   |
| ***** |      |       |    |    |   |   |    |   |   |
| *     | 1.   | /     | 5/ | *  | 0 | * | 0  | * | 0 |
| *     | 3.   | /     | 5/ | *  | 1 | * | 0  | * | 0 |
| *     | 4.   | /     | 4/ | *  | 1 | * | 0  | * | 1 |
| *     | 5.   | /     | 4/ | *  | 0 | * | 1  | * | 1 |
| *     | 6.   | /     | 5/ | *  | 0 | * | 1  | * | 0 |



# Primimplikāns lista

|    | IND. | INDL. | LI.PRIMIMPL.             |    | IND. | PRIMIMPL.                |
|----|------|-------|--------------------------|----|------|--------------------------|
| 1. | 5    | 5     | $\neg X, \neg Y, \neg Z$ |    |      |                          |
| 2. | 1    | 1     | $X, Y$                   |    |      |                          |
|    |      |       |                          | 3. | 5    | $\neg U, \neg X, \neg Y$ |
|    |      |       |                          | 4. | 4    | $\neg U, \neg X, Z$      |
|    |      |       |                          | 5. | 4    | $Y, Z$                   |
|    |      |       |                          | 6. | 5    | $\neg U, X, Y$           |

1.4 Két nem teljesen meghatározott, 6 db 1-pontot tartalmazó négy-  
változós függvényből álló függvényrendszer minimalizálása.

Felhasznált gépidő 18 mp.

$F_1 = -U \cdot -X \cdot -Y \cdot -Z + -U \cdot X \cdot Y \cdot Z + U \cdot -X \cdot -Y \cdot -Z + U \cdot X \cdot Y \cdot Z,$   
 $-F_1 = -U \cdot -X \cdot Y \cdot -Z + -U \cdot -X \cdot Y \cdot Z + -U \cdot X \cdot -Y \cdot Z + U \cdot -X \cdot Y \cdot -Z + U \cdot -X \cdot Y \cdot Z + U \cdot X \cdot -Y \cdot -Z + U \cdot X \cdot -Y \cdot Z,$   
 $F_2 = -U \cdot -X \cdot -Y \cdot Z + -U \cdot -X \cdot Y \cdot Z + -U \cdot X \cdot Y \cdot Z + U \cdot -X \cdot -Y \cdot -Z,$   
 $-F_2 = -U \cdot -X \cdot Y \cdot -Z + -U \cdot X \cdot -Y \cdot Z + U \cdot X \cdot Y \cdot -Z + U \cdot -X \cdot -Y \cdot Z + U \cdot -X \cdot Y \cdot -Z + U \cdot X \cdot -Y \cdot Z + U \cdot X \cdot Y \cdot -Z + U \cdot X \cdot Y \cdot -Z$

A feldolgozandó függvény  
beolvasott alakja:

# A feldolgozás eredménye:

## Diszjunkt halmazok listája

| SORSZ. | P IND. | P FUNK.I. | DT.E.SRSZ. | DT.ELEMET  |
|--------|--------|-----------|------------|------------|
| 1.     | 1      | 1         | 2          | U.X.Y.Z    |
| 2.     | 4      | 4         | 3          | -U.-X.-Y.Z |
| 3.     | 5      | 5         | 5          | -U.X.Y.Z   |
| 4.     | 4      | 4         | 6          | -U.-X.Y.Z  |

## A függvény adatai:

|       |        |                   |   |
|-------|--------|-------------------|---|
|       |        | FÜGGVÉNYEK SZÁMA: | 2 |
| INDEX | FV.NEV | VALTOZOK SZÁMA:   | 4 |
| 1     | F1     | 1-PONTOK SZÁMA:   | 5 |
| 1     | F1     |                   |   |
| 4     | F2     |                   |   |
| 4     | F2     |                   |   |

## Lefedési táblázat:

|       |      |   |       |    |   |    |   |   |    |   |
|-------|------|---|-------|----|---|----|---|---|----|---|
| ***** |      |   |       |    |   |    |   |   |    |   |
| *     | *    | * |       |    |   |    |   |   |    |   |
| *     | PONT | * | SORSZ | *  | 2 | *  | 3 | * | 5  | * |
| *     | *    | * | *     | *  |   | *  |   | * |    | * |
| ***** |      |   |       |    |   |    |   |   |    |   |
| *     | *    | * |       |    |   |    |   |   |    |   |
| *     | PRI  | * | IND   | *  | / | 1/ | * | / | 4/ | * |
| *     | *    | * | *     | *  |   |    | * |   |    | * |
| ***** |      |   |       |    |   |    |   |   |    |   |
| *     | 1.   | * | /     | 5/ | * | 0  | * | 0 | *  | 0 |
| *     | 2.   | * | /     | 1/ | * | 1  | * | 1 | *  | 0 |
| *     | 3.   | * | /     | 5/ | * | 0  | * | 0 | *  | 0 |
| *     | 4.   | * | /     | 4/ | * | 0  | * | 0 | *  | 1 |
| *     | 5.   | * | /     | 5/ | * | 0  | * | 0 | *  | 0 |
| *     | 6.   | * | /     | 4/ | * | 0  | * | 1 | *  | 1 |
| *     | 7.   | * | /     | 5/ | * | 0  | * | 1 | *  | 0 |
| *     | 8.   | * | /     | 5/ | * | 1  | * | 0 | *  | 0 |



# Primimplikāns lista

| IND. | INDL. | L.PRIMIMPL. | IND.     | PRIMIMPL. |
|------|-------|-------------|----------|-----------|
| 1.   | 5     | 5           | -X.-Y.-Z |           |
| 2.   |       |             | 1        | X.Y       |
| 3.   |       |             | 5        | -U.-X.-Y  |
| 4.   |       |             | 4        | -U.-X.Z   |
| 5.   |       |             | 5        | -U.-Y.-Z  |
| 6.   |       |             | 4        | Y.Z       |
| 7.   |       |             | 5        | -U.X.Y    |
| 8.   |       |             | 5        | X.Y.Z     |

1.5 Három nem teljesen meghatározott, 3 db 1-pontot tartalmazó négyváltozós függvényből álló függvényrendszer együttes egyszerűsítése.

Felhasznált gépidő 12 mp.

A feldolgozandó függvény beolvasott alakja:

```
T1=X1.-X2.-X3.-X4,
-T1=-X1.X2.X3.X4+X1.-X2.X3.-X4+X1.-X2.-X3.-X4+X1.X2.X3.-X4,
T2=X1.-X2.X3+X1.-X2.-X3.-X4,
-T2=-X1.X2.X3.X4,
T3=X1.-X2.X3+X1.-X2.-X3.-X4,
-T3=-X1.-X2.X3.-X4.
```

A feldolgozás eredménye:

### Diszjunkt halmazok listája

| SORSZ. | P IND. | P MUNK.I. | DT.E.SRSZ. | DT.ELEMEI      |
|--------|--------|-----------|------------|----------------|
| 1.     | 21     | 4         | 1          | X1.-X2.-X3.-X4 |
| 2.     | 20     | 4         | 2          | X1.-X2.X3.-X4  |
| 3.     | 20     | 4         | 3          | X1.-X2.X3.X4   |

### Primimplikáns lista

|    | IND. | INDL. | L.PRIMIMPL. |    | IND. | PRIMIMPL |
|----|------|-------|-------------|----|------|----------|
| 1. | 20   | 16    | X1          | 2. | 4    | -X2      |
|    |      |       |             | 3. | 20   | -X3      |
|    |      |       |             | 4. | 4    | -X4      |
| 5. | 21   | 1     | X1.-X3      |    |      |          |

### A függvény adatai

| INDEX | FV.NEV | FUGGVENYEK SZAMA: | 3 |
|-------|--------|-------------------|---|
| 1     | T1     | VALTOZOK SZAMA:   | 4 |
| 1     | T1     | 1-PONTOK SZAMA:   | 3 |
| 4     | T2     |                   |   |
| 4     | T2     |                   |   |
| 16    | T3     |                   |   |
| 16    | T3     |                   |   |

## Lefedési táblázat

|        |    |       |       |       |   |       |   |       |   |
|--------|----|-------|-------|-------|---|-------|---|-------|---|
| *****  |    |       |       |       |   |       |   |       |   |
| *      |    | *     |       | *     |   | *     |   | *     |   |
| * PONT | *  | SORSZ | *     | 1     | * | 2     | * | 3     | * |
| *      |    | *     |       | *     |   | *     |   | *     |   |
| *****  |    |       |       |       |   |       |   |       |   |
| *      |    | *     |       | *     |   | *     |   | *     |   |
| * PRI  | *  | IND   | *     | / 21/ | * | / 20/ | * | / 20/ | * |
| *      |    | *     |       | *     |   | *     |   | *     |   |
| *****  |    |       |       |       |   |       |   |       |   |
| *      | 1. | *     | / 20/ | *     | 1 | *     | 1 | *     | 1 |
| *      | 2. | *     | / 4/  | *     | 1 | *     | 1 | *     | 1 |
| *      | 3. | *     | / 20/ | *     | 1 | *     | 0 | *     | 0 |
| *      | 4. | *     | / 4/  | *     | 1 | *     | 1 | *     | 0 |
| *      | 5. | *     | / 21/ | *     | 1 | *     | 0 | *     | 0 |

1.6 Három nem teljesen meghatározott, 3 db 1-pontot tartalmazó négyváltozós függvényből álló függvényrendszer együttes minimalizálása.

Felhasznált gépidő 15 mp.

A feldolgozandó függvény beolvasott alakja:

$T1 = X1.-X2.-X3.-X4,$   
 $-T1 = -X1.X2.X3.X4 + X1.-X2.X3.-X4 + -X1.-X2.-X3.-X4 + X1.X2.X3.-X4,$   
 $T2 = X1.-X2.X3 + X1.-X2.-X3.-X4,$   
 $-T2 = -X1.X2.X3.X4,$   
 $T3 = X1.-X2.X3 + X1.-X2.-X3.-X4,$   
 $-T3 = -X1.-X2.X3.-X4,$   
 \*

### A feldolgozás eredménye:

#### Diszjunkt halmazok listája

| SORSZ. | P IND. | P MUNK.T. | DT.E.SRSZ. | DT.ELEMEI      |
|--------|--------|-----------|------------|----------------|
| 1.     | 21     | 4         | 1          | X1.-X2.-X3.-X4 |
| 2.     | 20     | 4         | 2          | X1.-X2.X3.-X4  |
| 3.     | 20     | 4         | 3          | X1.-X2.X3.X4   |



# Primimplikáns lista

|    | IND. | INDL. | L.PRIMIMPL. |    | IND. | PRIMIMPL |
|----|------|-------|-------------|----|------|----------|
| 1. | 20   | 16    | X1          | 2. | 4    | -X2      |
|    |      |       |             | 3. | 20   | -X3      |
|    |      |       |             | 4. | 4    | -X4      |
| 5. | 21   | 1     | X1.-X3      |    |      |          |

## A függvény adatai

| INDEX | FV.NEV | FÜGGVÉNYEK SZÁMA: |   |
|-------|--------|-------------------|---|
| 1     | F1     |                   | 3 |
| 1     | F1     | VALTOZOK SZÁMA:   | 4 |
| 4     | F2     | I-PONTOK SZÁMA:   | 3 |
| 4     | F2     |                   |   |
| 16    | F3     |                   |   |
| 16    | F3     |                   |   |

## Lefedési táblázat

|       |      |       |       |       |   |       |   |       |   |
|-------|------|-------|-------|-------|---|-------|---|-------|---|
| ***** |      |       |       |       |   |       |   |       |   |
| *     | *    | *     |       |       |   |       |   |       |   |
| *     | POIT | SORSZ | *     | 1     | * | 2     | * | 3     | * |
| *     | *    | *     |       |       |   |       |   |       |   |
| ***** |      |       |       |       |   |       |   |       |   |
| *     | PRI  | IND   | *     | / 21/ | * | / 20/ | * | / 20/ | * |
| *     | *    | *     |       |       |   |       |   |       |   |
| ***** |      |       |       |       |   |       |   |       |   |
| *     | 1.   | *     | / 20/ | *     | 1 | *     | 1 | *     | 1 |
| *     | 2.   | *     | / 4/  | *     | 1 | *     | 1 | *     | 1 |
| *     | 3.   | *     | / 20/ | *     | 1 | *     | 0 | *     | 0 |
| *     | 4.   | *     | / 4/  | *     | 1 | *     | 1 | *     | 0 |
| *     | 5.   | *     | / 21/ | *     | 1 | *     | 0 | *     | 0 |



PROGRAMRENDSZER BOOLE-FÜGGVÉNYRENDSZER EGYÜTTES EGYSZERÜSÍTÉSÉRE  
VAGY MINIMALIZÁLÁSÁRA

/Felhasználói leírás/

A programok a CDC 3300-as számítógépre FORTRAN nyelven készültek. Az alábbiakban két minimalizáló és egyszerűsítő programot ismertetünk. A két programot egy közös beolvasó és kiíró rész egészíti ki.

1. Minimalizáló program /FVMIN/

1.1 A program feladata

Diszjunktív normálformában /DNF/ megadott egyetlen teljesen meghatározott vagy nem teljesen meghatározott Boole-függvénynek, valamint több nem feltétlenül teljesen meghatározott függvénynek /függvényrendszernek/ az előforduló változók száma szerinti együttes minimalizálása ill. a minimalizáláshoz szükséges adatok szolgáltatása. Ezek az adatok a teljes primimplikáns lista, a lényeges primimplikánsok listája, a lényeges primimplikánsok által le nem fedett diszjunkt tartományok listája, és lefedési táblázat e diszjunkt halmazokra. Az  $F$  függvényrendszert komponens függvényein keresztül  $F = \{f_1, f_2, \dots, f_m\}$  adjuk meg. A komponens függvényeket pedig az  $f_{i1}$  és  $f_{i0}$  függvényekkel kell megadni az algoritmus számára.

$$f_{i1} = \begin{cases} 1 & \text{ha } f_i = 1 \\ 0 & \text{egyébként} \end{cases}$$

$$f_{i0} = \begin{cases} 1 & \text{ha } f_i = 0 \\ 0 & \text{egyébként} \end{cases}$$

A teljesen meghatározott  $f$  függvényre  $f_1 = f$  és  $f_0 = \bar{f}$ .

Szokás a nem teljesen meghatározott függvényt  $f_1$  és  $f$  -vel is megadni, ahol



$$f_{\phi} = \begin{cases} 1 & \text{ha } f \text{ nem meghatározott} \\ 0 & \text{egyébként} \end{cases}$$

$f_{\phi}$ -t szokás a meghatározatlansági tartomány karakterisztikus függvényének is nevezni.

## 1.2 A program szerkezete

A program három fő részből és egy kisegítő részből áll. Ezek a

- beolvasó program
- minimalizáló algoritmus
- az algoritmus eredményének kiértékelése és kiírása szélesnyomtatón
- bitműveleteket végző FORTRAN rutinok.

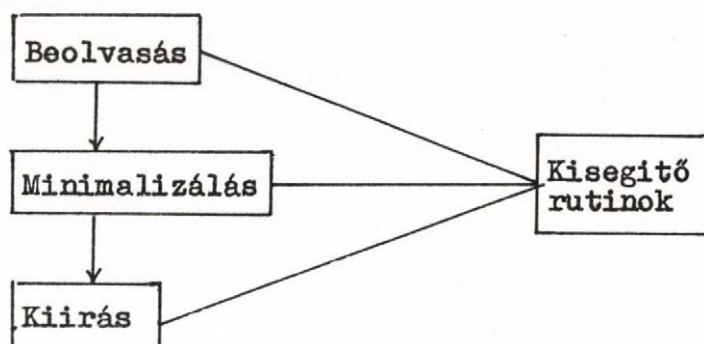
A beolvasó program a függvény DNF-ját beolvassa, és szintaktikailag ellenőrzi azt. Hiba esetén hibajelzést ad, és a szintaktikai vizsgálat után nem folytatja a munkát. Ha a formula helyes volt, a függvény DNF-jának egy olyan belső gépi reprezentációját állítja elő, amely a minimalizáló algoritmus bemenő adata.

A minimalizáló algoritmus a gráf módszer [1] alapján felírja az összes primimplikánst vagy együttes primimplikánst, és megjelöli a lényegeseket. A függvény pontjait a primimplikánsokkal való lefedés alapján ekvivalencia-osztályokba osztja /diszjunkt halmazok/ és megjelöli azokat a diszjunkt halmazokat, amelyeket lefed lényeges primimplikáns.

A kiértékelő és kiíró programrész kinyomtatja a minimalizált függvények neveit, a változók számát, a teljes primimplikáns listát, megjelölve a lényeges primimplikánsokat és egy lefedési táblázatot a lényeges primimplikánsok által le nem fedett diszjunkt tartományokra, és a nem lényeges primimplikánsokra.

A kisegítő rutinok, amelyeket az előző három programrész használ fel, a gépi szó tartalmának bitenkénti kezelését teszik lehetővé. Ezek a rutinok egy vagy két gépi szóra elvégzik bitenként a szokásos logikai műveleteket /negáció, diszjunkció, konjunkció stb./

A program sémája:



1. ábra

### 1.3 Adatelőkészítés

A beolvasó program alapfunkciója a DNF gépi reprezentációjának előkészítése. A függvény diszjunktív normálformájának szokásos alakját minimális mértékben kell csak átalakítani /linearizálni/, hogy a beolvasó számára érthető legyen.

A formulában felhasznált jelek a következők:

|                      |   |   |
|----------------------|---|---|
| diszjunkció jele:    | + |   |
| konjunkció jele:     | . |   |
| negáció jele:        | - |   |
| "függvény vége" jel: | , |   |
| "adat vége" jel:     | ⌘ |   |
| változónév           | } | betűvel kezdődő, legfeljebb négyjegyű<br>alfanumerikus karaktersorozat lehet. |
| függvénynév          |   |   |

Pl. ha  $f_1 = a_1 \wedge b_1 \wedge \bar{c} \vee \bar{a}_1$  és  $hx_2 = x \vee \bar{y}_2$  formulákat kell megadni a program bemenő adataként, akkor a formulák bemenő adattá való átírásának eredménye:  $f1=a1.b1.-c+-a1$ ,  $hx2=x+-y2$  ⌘

A formula felírásában a space jeleknek nincs jelentőségük.

A beolvasó program a beolvasott függvények DNF-i belső reprezentációjából a minimalizáló program bemenő adatát elkészíti az alábbi esetekben.



a. Egyetlen teljesen meghatározott függvény esetén

Ekkor az egy teljesen meghatározott függvényt minimalizáló speciális algoritmus számára előállítja a függvény kitüntetett diszjunktív normálformáját a függvény valamely tetszőleges DNF-ja alapján.

A bemenő adat a függvény tetszőleges DNF-ja, azaz  
 $f_{10} = a.b5.-d6+d6.-a \equiv$

b. Nem teljesen meghatározott függvény vagy függvényrendszer esetén

A függvényrendszert /egyetlen függvényből állót is/ minimalizáló program bemenő adata a komponens függvényeket megadó  $f_{11}, f_{10}$  függvények kitüntetett diszjunktív normálformái. E függvények kötelező jelölése a beolvasó program számára  $f_{11}=f_i$ ,  $f_{10}=-f_i$ . Az említett bemenő adatot a beolvasó program a következő három esetben tudja előállítani.

b-1. Megadjuk egyes komponensfüggvényekhez tartozó  $f_{11}, f_{10}$  tetszőleges DNF-it bármilyen sorrendben.

Pl. ha  $fa_1 = d \wedge \bar{a} \wedge \bar{b} \wedge \bar{c} \vee \bar{a} \wedge b \wedge c \wedge d$ ,

$fa_0 = a \wedge b \wedge \bar{c} \wedge \bar{d} \vee \bar{a} \wedge \bar{c} \wedge b \wedge d$ ,

$fb_1 = d \wedge \bar{a} \wedge \bar{b} \wedge \bar{c}$ ,  $fb_0 = b \wedge \bar{a} \wedge \bar{c} \wedge \bar{d} \vee a \wedge c \wedge \bar{b} \wedge \bar{d}$

az egyszerűsítendő függvények, akkor a program egy bemenő adata a következő lehet:

$fa = d.-a.-b.-c+.-a.b.c.d$ ,  $-fa = a.b.-c.-d+.-a.-c.b.d$ ,

$fb = d.-a.-b.-c$ ,  $-fb = b.-a.-c.-d.+a.c.-b.-d \equiv$

b-2. Az  $f_{j1}, f_{j0}$ -al meg nem adott komponens függvények mind teljesen meghatározottak, és  $f_i$  vagy  $\bar{f}_i$  közül legalább az egyik ismert. Ebben az esetben meg kell adni az ismert függvények tetszőleges DNF-it tetszőleges sorrendben, és utolsó függvényként egy, a már felsorolt függvényekben szereplő változók konjunkcióját, azaz egy  $S = \bigwedge_{i=1}^K v_i$  formulát, ahol S olyan változónév, amely nem szerepel komponens függvény neveként, K pedig a komponens függvényekben előforduló változók száma. Pl.  $F = \{f_1, f_2, f_3, f_4\}$  esetén, ha adott  $f_1, \bar{f}_1, f_2, \bar{f}_3, f_{4,1}, f_{4,0}$ , akkor a bemenő adatok



$-f40 = -c.p.t + -t.-c.-a + -p.-c,$   
 $f2 = -a + b.p + -b.t + p.t, \quad -f3 = a.c.p.-n + a.-c,$   
 $f1 = a.-t.b + -a.b, \quad f41 = c.-t, \quad -f1 = -b + a.t,$   
 $g = a.b.c.p.t.-n \times$

b-3. Az  $f_1, \bar{f}_1$  vagy  $f_{11}, f_{10}$  alakban pontosan meg nem adott komponens függvények g meghatározatlansági tartománya közös. Ekkor meg kell adni az ismert függvények tetszőleges DNF-it tetszőleges sorrendben, és utolsó függvényként a g függvény DNF-ját, de  $-g = \dots$  alakban.

Pl.  $F = \{f_1, f_2, f_3, f_4, f_5\}$ , ahol  $f_1$  teljesen meghatározott és  $f_1, \bar{f}_1$  adott,  $f_2$  nem teljesen meghatározott, de  $f_{21}, f_{20}$  adott,  $f_3, f_4, f_5$  meghatározatlansági tartománya, amelynek karakterisztikus függvénye g, közös és adott  $g, f_{31}, f_{41}, f_{50}$ . A beolvasó program egy lehetséges bemenő adata:

$f1 = d.-e.f + -d.f, \quad -f1 = -f + d.e,$   
 $f2 = f.-e, \quad -f2 = -f.d.e + -e.-f.-a + -d.-f,$   
 $f3 = -d.e.-f.a.z + d.e.-f.-z,$   
 $f4 = -d.-e.-f.a + -d.e.-f.z.a,$   
 $-f5 = d.-f.-e.z.a + d.-e.f,$   
 $-g = d.e.f + -d.-e.f + -d.e.f \times$

Az adatokat lyukkártyán kell rögzíteni. A lyukkártya 1-80 oszlopai tetszés szerint felhasználhatók, mivel a space jelnek nincs jelentősége /lásd 2. ábra/. Uj sorra való áttérés mindig elválasztó jelnél (=,+,.,",") történhet. Folytatásjel nincs.

#### 1.4 Hibaüzenetek

##### A beolvasó program hibaüzenetei:

Ha a beolvasó program a vizsgálat során szintaktikai hibát talál, a következő hibaüzenetek valamelyike kerül kiírásra.

##### 1. A KARAKTER NEM FUEGGVEENYNEEV

A KARAKTER INDEXE A LISTAABAN: IND

A KARAKTER BELSOE NEVE: L IND

A FUEGGVEENYNEEV SORSZAAMA: KS

##### 2. A FUEGGVEENYNEEVET NEM KOEVETI EGYENLOESEEGJEL

továbbá az 1. alatti üzenet 2.-4. sora.

3. ILLEGAALIS ELVAALASZTOJEL A LISTAABAN

Az 1. alatti üzenet 2. és 4. sora

4. FELESLEGES VEEGZAARJEL

Az 1. alatti üzenet 2. és 4. sora

5. A KARAKTER AZ OET MEGELOEZOE KARAKTER UTAAN NEM KOEVETKEZHET

Továbbá az 1. alatti üzenet 2.-4. sora

6. BE NEM FEJEZETT ZAARJELPAAROK A FUEGGVEENYBEN

A ZAARJELPAAROK SZÁMA: IZAR

A FUEGGVEENY SORSZÁMA: KS

7. A KARAKTERT MEGELOEZOE MAASODIK KARAKTER ILLEGAALIS

Továbbá az 1. alatti üzenet 2.-4. sora

8. A KARAKTERT KOEVETOE KARAKTER ILLEGAALIS

Továbbá az 1. alatti üzenet 2.-4. sora

A függvény sorszáma megadja, hogy a függvény hányadik a vizsgálat alá vont függvények sorozatában.

A szintaktikai vizsgálat végén, ha a formulában hibát talált, akkor a - SZINTAKTIKAI HIBAK - hibaüzenetet írja ki.

PONTOK SZÁMA GE 500

500-NAL TOEBB 1-PONT ATIRT-BAN

500-NAL TOEBB 0-PONT ATIRT-BAN

A minimalizáló rész hibaüzenetei:

-  $f_{11}, f_{10}$  ellentmondásossága esetén:

ELLENTMONDAS A 0- ÉS 1-PONTOK KÖZÖTT

Az i-DIK 1-PONT ES A j-DIK 0-PONT EGYFORMA

- Ha a primimplikánsok száma nagyobb 140-nél:

TULNAGY MPR

- Ha a kombinációk képzésénél 200-nál több kombináció áll elő:

$MPR=i_1, IPR=i_2, NW=i_3$

Szintaktikai hibák esetén a hibaüzenet után tovább fut a program, és a teljes formulára elvégzi a szintaktikai vizsgálatot, majd a "SZINTAKTIKAI HIBAK" kiírása után megáll. A többi hibaüzenet esetén a program végrehajtása azonnal felfüggesztődik.

### 1.5 A program eredménye

A minimalizáló program eredményét a következő módon kapjuk meg:

a/ A feldolgozott függvények és a hozzájuk tartozó  $2^i$  alakú indexek listájának és a függvények, és a változók számának kiírása az alábbi formában.

| INDEX | FV. NÉV |
|-------|---------|
| 1     | FK1     |
| 2     | G5      |
| 32    | Y       |

FÜGGVÉNYEK SZÁMA: 3

VÁLTOZÓK SZÁMA: 7

b/ A Boole-függvény vagy függvények összes primimplikánsainak és lényeges primimplikánsainak listáját. Több függvény együttes egyszerűsítése esetén ez a lista egy mutatószám listával bővül, amely jelzi, hogy az illető primimplikáns mely függvényeknek primimplikánsa ill. lényeges primimplikánsa.

A lista formája:

|    | IND       | L.PR.IMPL. | IND.  | PR.IMPL.  |
|----|-----------|------------|-------|-----------|
| 1. | $n_1 k_1$ | a.-b.c     | -     | -         |
| 2. |           |            | $n_2$ | -a.c.d.-b |
| 3. | ...       |            |       |           |
| :  |           |            |       |           |

$n_1$  azon függvények indexeinek összege, amelyeknek az illető primimplikáns implikánsa.

$k_1$  azon függvények indexeinek összege, amelyeknek az illető primimplikáns lényeges primimplikánsa.



c/ A lényeges primimplikánsok által le nem fedett függvénypontok diszjunkt halmazainak listája, és egy index lista, amely egyrészt azt mutatja, hogy az egyes diszjunkt halmazok mely függvényeknek pontjai /F.IND./, másrészt pedig azt, hogy a lényeges primimplikánsok mely függvényekben nem fedik le a diszjunkt halmazt /F.MUNK.I./.

A lista:

| SORSZ. | DT.E.SORSZ. | DT.ELEMEI | F.IND. | F.MUNK.I. |
|--------|-------------|-----------|--------|-----------|
| 1.     | $i_1$       | $m_1$     | $j_1$  | $k_1$     |
|        | $i_2$       | $m_2$     | .      | .         |
|        | .           | .         | .      | .         |
|        | $i_k$       | $m_k$     |        |           |
| 2. ... |             |           |        |           |
| ⋮      |             |           |        |           |

d/ Lefedési táblázat, amely megmutatja, hogy az egyes diszjunkt halmazokat mely primimplikánsok fedik le.

A táblázat:

| PONT. | SORSZ. | $n_1$ | $n_2$ | ... |
|-------|--------|-------|-------|-----|
| PRI.  | IND.   | $j_1$ | $j_2$ |     |
| $k_1$ | $i_1$  | 1     | 1     |     |
| $k_2$ | $i_2$  | 0     | 1     |     |
| ⋮     |        |       |       |     |

ahol  $n_1$  a diszjunkt halmaz első elemének sorszáma,  $j_1$  a diszjunkt halmaz indexe,  $k_j$  a primimplikáns sorszáma a listában,  $i_j$  a primimplikáns indexe.

Megjegyzés: A program max. 23 változós függvényekre alkalmazható.  
A függvényrendszer komponenseinek száma max. 23.

## 1.6 Futtatási utasítás

A programfutáshoz a 7-es vagy a 8-as disc-re van szükség. A program bináris alakja a 8-as discen két file-n található.

File nevek: MABEOLV, FUGGVENY MINIMALIZALAS

A 8-as discen lévő MABEOLV, FUGGVENY MINIMALIZALAS file-k megnyitása után a megfelelő dsi-k egyikét a TASKNAME kártyán, a másikat egy BIN kártyán megadjuk. A BIN kártya és az "END of JOB" kártya közé adatként elhelyezzük a feldolgozandó függvény, vagy függvények tetszőleges diszjunktív normálformáit.

```
Pl. $JOB,szám,név,idő,sorok száma,, ,megjegyzés
    $SCHED,CORE=70,SCR=16,854=1,CLASS=B
    $DEF 0,W,MIN,222020,FUGGVENY MINIMALIZALAS,01,,0
    $DEF 0,W,MOLV,222020,MABEOLV,01,,0
    $X,MIN
    $BIN,MOLV,
```

Adatkártyák

EOJ kártya

A program FORTRAN forrásnyelvi alakját a 7-es discen találjuk.

File nevek: BEOLV1 és MINIMALIZALAS.

A futtatáshoz /ha a futásközbeni ellenőrző kiírásokat is meg akarjuk kapni/ a következő JOB-ot kell összeállítani.

```
$JOB,szám,név,idő,sorok száma,, ,megjegyzés
$SCHED,CLASS=B,CORE=80,SCR=20,854=1
$DEF 0,W,BFV,222020,MINIMALIZALAS,01,,0
$DEF 0,W,BE1,21C111,BEOLV1,01,,0
$COSY
```

|         |       |                |
|---------|-------|----------------|
| BITMUV  | DECK/ | I=BFV,H,G=FTNU |
| OLVAS1  | DECK/ | I=BE1,H,G=FTNU |
| MINFO   | DECK/ | I=BFV,H,G=FTNU |
| TABLAZ  | DECK/ | I=BFV,H,G=FTNU |
| SORREND | DECK/ | I=BFV,H,G=FTNU |
| MINIMAL | DECK/ | I=BFV,H,G=FTNU |
| IR      | DECK/ | I=BFV,H,G=FTNU |
| ALTMIN  | DECK/ | I=BFV,H,G=FTNU |
| EGYMIN  | DECK/ | I=BFV,H,G=FTNU |

```

ENDCOSY/
$FTNU I=SHO,X,L,N,D,S
$X,LGO
      Adatkártyák
EOJ kártya

```

## 2. Egyszerűsítő program /FV.RED/

### 2.1 A program feladata

Diszjunktív normálformában /DNF/ megadott egyetlen teljesen meghatározott vagy nem teljesen meghatározott Boole-függvénynek, valamint több nem feltétlenül teljesen meghatározott függvénynek /függvényrendszernek/ az előforduló változók száma szerinti együttes egyszerűsítése ill. az egyszerűsítéshez szükséges adatok szolgáltatása.

Ezek az adatok a teljes primimplikáns lista, a lényeges primimplikánsok listája, a lényeges primimplikánsok által le nem fedett diszjunkt tartományok listája és lefedési táblázat e diszjunkt halmazokra. Az F függvényrendszert komponens függvényein keresztül  $F = \{f_1, f_2, \dots, f_m\}$  adjuk meg. A komponens függvényeket pedig az  $f_{11}$  és  $f_{10}$  függvényekkel kell megadni az algoritmus számára.

### 2.2 A program szerkezete

A program szerkezetileg az 1.-ben leírt FV.MIN programhoz hasonlít. A program három fő részből és egy kisegítő részből áll.

Ezek a

- beolvasó program
- egyszerűsítő algoritmus
- az algoritmus eredményének kiértékelése és kiírása szélesnyomtatón
- bitműveleteket végző FORTRAN rutinok.

A FV.MIN és FV.RED programokhoz közös a beolvasó program. Ennek ismertetése 1.2.-ben található.



### 2.3 Adatelőkészítés

Azonos az 1.3.-ban leírtakkal.

### 2.4 Hibaüzenetek

Leírását lásd 1.4.-ben.

### 2.5 A program eredménye

Az egyszerűsítő program eredményét ugyanolyan formában kapjuk meg, mint ahogy azt az 1.5.-ben leírtuk. A különbség annyi, hogy nem feltétlenül a teljes primimplikáns listát kapjuk meg.

### 2.6 Futtatási utasítás

A programfutáshoz a 7-es vagy a 8-as disc-re van szükség. A program bináris alakja a 8-as discen két file-n található.

File nevek: MABEOLV, BOOLE.

A 8-as discen lévő MABEOLV, BOOLE file-k megnyitása után a megfelelő diszk egyikét a TASKNAME kártyán, a másikat egy BIN kártyán megadjuk. A BIN kártya és az "END of JOB" kártya közé adatként elhelyezzük a feldolgozandó függvény vagy függvények tetszőleges diszjunktív normálformáit.

Pl. \$JOB,szám,név,idő,sorok száma,, ,megjegyzés

\$SCHED,CORE=70,SOR=16,854=1,CLASS=B

\$#DEF 0,W,RED,222020,BOOLE,01,,0

\$#DEF 0,W,MOLV,222020,MABEOLV,01,,0

\$X,RED

\$BIN,MOLV

Adatkártyák

EOJ kártya

A program FORTRAN forrásnyelvi alakját a 7-es discen találjuk.

File nevek: BEOLV1 és MINIMALIZALAS.

A futtatáshoz /ha a futásközbeni kiírásokat is meg akarjuk kapni/

következő JOB-ot kell összeállítani.

P1. \$JOB,szám,név,idő,sorok száma,, ,megjegyzés

\$SCHED,CORE=70,SCR=6,854=1,CLASS=B

\$DEF 0,W,BFV,222020,MINIMALIZALAS,01,,0

\$DEF 0,W,BE1,21C111,BEOLV1,01,,0

\$COSY

NEMRED DECK/ I=BFV,H,G=FTNU

EGYSZ DECK/ I=BFV,H,G=FTNU

TABLAZ DECK/ I=BFV,H,G=FTNU

SORREND DECK/ I=BFV,H,G=FTNU

ALTF DECK/ I=BFV,H,G=FTNU

EGYF DECK/ I=BFV,H,G=FTNU

IR DECK/ I=BFV,H,G=FTNU

BITMUV DECK/ I=BFV,H,G=FTNU

OLVS1 DECK/ I=BE1,H,G=FTNU

ENDCOSY/

\$FTNU I=SHO,L,S,X

\$X,LGO

Adatkártyák

EOJ kártya

## TARTALOMJEGYZÉK

|   |    |
|---|----|
| ÖSSZEFOGLALÁS .....                                   | 3  |
| 1. MINIMALIZÁLÓ ÉS EGYSZERÜSÍTŐ PROGRAMRENDSZER ..... | 9  |
| 1.1 A program feladata .....                          | 9  |
| 1.2 A program szerkezete .....                        | 10 |
| 2. A PROGRAM RÉSZLETES LEÍRÁSA .....                  | 13 |
| 2.1 A beolvasó program .....                          | 13 |
| 2.2 A program által végrehajtandó feladatok .....     | 14 |
| 2.3 A feladatsorozat végrehajtásának módja .....      | 20 |
| 2.4 A megvalósított program ismertetése .....         | 21 |
| 3. MINIMALIZÁLÓ ILL. EGYSZERÜSÍTŐ PROGRAMRÉSZ .....   | 35 |
| 4. KIÍRÓ ILL. KIÉRTÉKELŐ PROGRAMRÉSZ .....            | 51 |
| IRODALOMJEGYZÉK .....                                 | 55 |
| MELLÉKLETEK .....                                     | 57 |
| 1. Melléklet .....                                    | 59 |
| 2. Melléklet .....                                    | 77 |



A TANULMÁNYOK sorozatban eddig megjelentek:

- 1/1973 Pásztor Katalin: Módszerek Boole-függvények minimális vagy nem redundáns,  $\{\wedge, \vee, \neg\}$  vagy  $\{\text{NOR}\}$  vagy  $\{\text{NAND}\}$  bázisbeli, zárójeles vagy zárójel nélküli formuláinak előállítására
- 2/1973 Вашкеви Иштван: Расчленение многосвязных промышленных процессов с помощью вычислительных машин
- 3/1973 Ádám György: A számítógépipar helyzete 1972 második felében
- 4/1973 Bányász Csilla: Identification in the Presence of Drift
- 5/1973\* Gyürki J.-Laufer J.-Grint M.-Somló J.: Optimalizáló adaptív szerszámgépirányítási rendszerek
- 6/1973 Szelke E. - Tóth K.: Felhasználói Kézikönyv /USER MANUAL/ a Folytonos Rendszerek Szimulációjára készült ANDISIM programnyelvhez
- 7/1973 Legendi Tamás: A CHANGE nyelv/multiprocesszor
- 8/1973 Klafszky Emil: Geometriai programozás és néhány alkalmazása
- 9/1973 R.Narasimhan: Picture Processing Using Pax
- 10/1973 Dibuz Á.-Gáspár J.-Várszegi S.: MANU-WRAP hátulphuzalozó, MSI-TESTER integrált áramköröket mérő, TESTOMAT-C logikai hálózatokat vizsgáló berendezések ismertetése
- 11/1973 Matolcsi Tamás: Az optimum-számítás egy új módszeréről
- 12/1973 Makroprocesszorok, programozási nyelvek. Cikkgyűjtemény az NJSzT és SzTAKI közös kiadásában. Szerkesztette: Legendi Tamás
- 13/1973 Jedlovsky Pál: Új módszer bonyolult rektifikáló oszlopok vegyész-mérnöki számítására
- 14/1973 Bakó András: MTA kutatóintézeteinek bérszámfejtése számítógéppel
- 15/1973 Ádám György: Kelet-nyugati kapcsolatok a számítógépiparban
- 16/1973 Fidrich I. - Uzsoy M.: LIDI-72 listakezelő rendszer a Digitális Osztályon, 1972.évi változat

- 17/1974 Gyurki József: Adaptív termelésprogramozó rendszer /APS/ termelőműhelyek irányítására
- 18/1974 Pikler Gyula: MINI-számítógépes interaktív alkatrész-program-író rendszer NC szerszámgépek automatikus programozásához
- 19/1974 Gertler, J.-Sedlak, J.: Software for process control
- 20/1974 Vámos, T.-Vassy, Z.: Industrial Pattern Recognition Experiment - A Syntax Aided Approach
- 21/1974 A KGST I.-15-1.: "Diszkrét rendszerek automatikus vezérlése" c. témában 1973. februárban rendezett szeminárium előadásai
- 22/1974 Arató, M.-Benczur, A.-Krámli, A.-Pergel, J.: Stochastic Processes, Part I.
- 23/1974 Benkó S.-Renner G.: Erősen telített mágneskörök számítógépes tervezési módszere
- 24/1974 Kovács György-Franta Lászlóné: Programcsomag elektronikus be-rendezések hátlaphuzalozásának tervezésére
- 25/1974 Járdán R. Kálmán: Háromfázisu tirisztoros inverterek állandósult tranziens jelenségei és belső impedanciája
- 26/1974 Gergely József: Numerikus módszerek sparse mátrixokra
- 27/1974 Somló János: Analitikus optimalizálás
- 28/1974 Vámos Tibor: Tárgyfelismerési kísérlet nyelvi módszerekkel
- 29/1974 Móricz Péter: Vegyészmérnöki számítási módszerek fázisegyensúlyok és kémiai egyensúlyok vizsgálatára
- 30/1974 Vassy, Z.-Vámos, T.: The Budapest Robot - Pragmatic Intelligence
- 31/1974 Nagy István: Frekvenciaosztásos középfrekvenciás inverter elmélete
- 32/1975 Singer D., Borossay Gy., Koltai T.: Gázhálózatok optimális irányítása különös tekintettel a Fővárosi Gázművek hálózataira
- 33/1975 Vámos, T.-Vassy, Z.: Limited and Pragmatic Robot Intelligence  
Mérő, L.-Vassy, Z.: A Simplified and Fastened Version of the Hueckel Operator for Finding Optimal Edges in Pictures



Галло В.: Программа для распознавания геометрических образов, основанная на лингвистическом методе описания и анализа геометрических структур

- 34/1975 László Nemes: Pattern Identification Method for Industrial Robots by Extracting the Main Features of Objects
- 35/1975 Garádi-Krámli-Ratkó-Ruda: Statisztikai és számítástechnikai módszerek alkalmazása kórházi morbiditás vizsgálatokban
- 36/1975 Renner Gábor: Elektromágneses tér számítása nagyhőmérsékletű anyagban
- 37/1975 Edgardo Felipe: Specification problems of a process control display
- 38/1975 Hajnal Andrásné: Nemlineáris egyenletrendszerek megoldási módszerei
- 39/1975\* A.Abd.El-Sattar: Control of induction motor by ....
- 40/1975 Gerhardt Géza: QDT Grafikus interaktív szubrutinok a CDC 3300 -GD'71 grafikus konfigurációra
- 41/1975 Arató M.-Benczur, A.-Krámli, A.-Pergel, J.: Stochastic Processes, Part II.
- 42/1975 Arató Mátyás: Fejezetek a matematikai statisztikából számítógépes alkalmazásokkal

---

A \* -gal jelölt kivételével a sorozat kötetei megrendelhetők az Intézet könyvtáránál /Budapest, I., Victor Hugo u. 18-22./.





